ENCE & TECHNOLOGY:

DESIGN ALTERNATIVES FOR COMPUTER NETWORK SECURITY

NBS Special Publication 500-21, Volume 1
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau consists of the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Institute for Computer Sciences and Technology, the Office for Information Programs, and the Office of Experimental Technology Incentives Program.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of the Office of Measurement Services, and the following center and divisions:

Applied Mathematics — Electricity — Mechanics — Heat — Optical Physics — Center for Radiation Research — Laboratory Astrophysics[2] — Cryogenics[2] — Electromagnetics[2] — Time and Frequency[2].

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement, standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; and develops, produces, and distributes standard reference materials. The Institute consists of the Office of Standard Reference Materials, the Office of Air and Water Measurement, and the following divisions:

Analytical Chemistry — Polymers — Metallurgy — Inorganic Materials — Reactor Radiation — Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services developing and promoting the use of available technology; cooperates with public and private organizations in developing technological standards, codes, and test methods; and provides technical advice services, and information to Government agencies and the public. The Institute consists of the following divisions and centers:

Standards Application and Analysis — Electronic Technology — Center for Consumer Product Technology: Product Systems Analysis; Product Engineering — Center for Building Technology: Structures, Materials, and Safety; Building Environment; Technical Evaluation and Application — Center for Fire Research: Fire Science; Fire Safety Engineering.

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in improving cost effectiveness in the conduct of their programs through the selection, acquisition, and effective utilization of automatic data processing equipment; and serves as the principal focus wthin the executive branch for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Institute consist of the following divisions:

Computer Services — Systems and Software — Computer Systems Engineering — Information Technology.

THE OFFICE OF EXPERIMENTAL TECHNOLOGY INCENTIVES PROGRAM seeks to affect public policy and process to facilitate technological change in the private sector by examining and experimenting with Government policies and practices in order to identify and remove Government-related barriers and to correct inherent market imperfections that impede the innovation process.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System; provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data — Office of Information Activities — Office of Technical Publications — Library — Office of International Standards — Office of International Relations.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.

[2] Located at Boulder, Colorado 80302.

COMPUTER SCIENCE & TECHNOLOGY:

Design Alternatives for Computer Network Security

Gerald D. Cole,

System Development Corporation
2500 Colorado Avenue
Santa Monica, CA 90406

Dennis K. Branstad, Editor

Systems and Software Division
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

Sponsored by the

Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Dr. Sidney Harman, Under Secretary

Jordan J. Baruch, Assistant Secretary for Science and Technology

U S NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Acting Director

Issued January 1978

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

National Bureau of Standards Special Publication 500-21, Volume 1
Nat. Bur. Stand. (U.S.), Spec. Publ. 500-21, Vol. 1, 173 pages (Jan. 1978)
CODEN: XNBSAV

Library of Congress Cataloging in Publication Data

Cole, Gerald D.

Design alternatives for computer network security.

(Computer science and technology) (NBS special publication ; 500-21, v. 1)

Supt. of Docs. no.: C13.10:500-21, v. 1.

1. Computers--Access control. 2. Computer networks--Security measures. I. Branstad, Dennis K. II. United States. National Bureau of Standards. Institute for Computer Sciences and Technology. III. Title. IV. Series. V. Series: United States. National Bureau of Standards. Special publication ; 500-21, v. 1.

QC100.U57 no. 500-21, vol. 1 [QA76.9.A25] 602'.1s [658.4'7] 77-608320

U.S. GOVERNMENT PRINTING OFFICE

WASHINGTON: 1978

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
(Order by SD Catalog No. C13.10:500-21, Vol. 1), Stock No. 003-003-01881-3 Price $6 per 2 volume set; sold in sets only
(Add 25 percent additional for other than U.S. mailing).

PREFACE

This publication was originally prepared for the Department of Defense under Contract Number DAAB03-73-C-1488 by the System Development Corporation in 1974. It has been revised for publication by the National Bureau of Standards under Contract Number 5-35934. The author of the paper was assisted in its development by members of the System Development Corporation's Systems Security Department including K. Auerback, J. Garwick, H. Grycner, D. Kaufman and the Department Head, C. Weissman. The editor has been assisted by Mrs. Gloria Bolotsky of the Systems and Software Division as well as others within NBS and DOD.

This document was originally prepared under the direction of the Department of Defense. Consequently, some of the nomenclature used is DOD oriented, e.g., classification levels mean Top Secret, Secret, Confidential and Unclassified. Rather than modify this nomenclature throughout the document, the editor requests that the reader adapt the concept of classification levels and protection levels to those accepted in any particular application. Sensitivity level may be substituted throughout the document for classification level if this concept is better defined and accepted.

The terms Security Controller, Network Security Controller, Cryptographic Controller, Key Distribution Center, Network Access Controller and Network Security Center have all been used in the literature to describe the same concept. This concept involves the use of a dedicated computer to control access to a computer network through the control of data encryption keys. An encryption key is a parameter, typically a binary number, that controls the processes of enciphering (encrypting) and deciphering computer data. An authorized user or terminal in a computer network is issued an encryption key to obtain access after the credentials of the user or terminal have been verified. In practice the concept of a security controller incorporates the use of a special process (program) in a computer, a special machine or a number of special machines (mini- or micro-computers) to control the security of the network through the generation and distribution of encryption keys.

A companion publication entitled "THE NETWORK SECURITY CENTER: A SYSTEM LEVEL APPROACH TO COMPUTER NETWORK SECURITY" is the result of the second phase of the NBS computer network security project. The two publications should be read as a series with the understanding that they were developed about two years apart (1974 and 1976). In addition a significant amount of research and development has been done in this area by NBS, SDC and others after these two reports were developed and a great deal of work is still going on. The reader is therefore cautioned that the results contained in these publications are not complete and any recommendations contained in them should not be accepted without further investigation into present developments.

NOTE

Certain commercial products are identified in this paper in order to specify adequately the experimental procedure, or to cite relevant examples. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the products or equipment identified are necessarily the best available for the purpose.

TABLE OF CONTENTS

Section		Page
1.	INTRODUCTION.	1
2.	NETWORK SECURITY POLICY AND REQUIREMENTS ISSUES	10
2.1	IDENTIFICATION/AUTHENTICATION ISSUES.	11
2.1.1	Authentication of Persons and Devices	11
2.1.2	Process and Host Level Authentication	12
2.1.3	Distributed Versus Centralized Authentication Checking. . .	13
2.1.4	N-th Party Authentication	14
2.2	ACCESS REQUEST/AUTHORIZATION ISSUES	15
2.2.1	Access Authorization Design Principles.	18
2.2.2	Authorization Checking at Local and Remote Nodes.	19
2.2.3	Composite Authorizations.	20
2.2.4	N-th Party Authorization.	21
2.2.5	Checking Required to Access the Authorization Mechanism . .	22
2.3	ACCESS CONTROL; ISSUES RELATED TO THE ESTABLISHMENT OF CONNECTIONS	24
2.3.1	Host Acceptance of Connections.	25
2.3.2	Profile Information to be Sent at Connection Establishment.	25
2.3.3	Error Handling.	26
2.4	ACCESS CONTROL; POLICY ISSUES AND REQUIREMENTS RELATED TO USE OF A CONNECTION	26
2.4.1	Security Control Mechanism as Part of the Communication Path.	26
2.4.2	Control Via Encryption Devices.	27
2.4.3	Degradation Due to Security Mechanisms.	28
2.4.4	Separation of Data and Control.	28
2.5	SECURITY MONITORING ISSUES.	31
2.5.1	Collection of Audit Information	31
2.5.2	A Network Security Center	31

TABLE OF CONTENTS

Section		Page
2.6	SECURITY ASSURANCE ISSUES.	32
2.6.1	Accreditation of the Security Mechanisms	33
2.6.2	Sufficiency of Protection.	33
2.6.3	Secrecy of the Mechanism Designs	34
2.6.4	Reliability and Failure Modes.	35
2.6.5	Self-Checking.	35
2.6.6	Interface to Physical and Procedural Controls.	35
2.7	OTHER POLICY AND REQUIREMENTS ISSUES	36
2.7.1	The User Interface to the Network.	36
2.7.2	Network Management	37
2.7.3	Meeting Network Traffic Needs.	38
2.7.4	Separation of Data Processing and Data Communications. . .	38
3.	NETWORK SECURITY MECHANISMS AT THE SECURITY CONTROLLER/ HOST LEVEL	40
3.1	IDENTIFICATION/AUTHENTICATION.	43
3.1.1	Identification Information	43
3.1.2	Providing Network-Wide Authentication.	44
3.1.3	The SC as a Pre-Connection Check or a Reference Check. . .	46
3.1.4	SC-to-SC Authentication.	46
3.1.5	The SC Role in N-th Party Authentication	47
3.2	ACCESS REQUEST/AUTHORIZATION	49
3.2.1	The Content of the Access Tables	49
3.2.2	Organization of the Table.	50
3.2.3	Usage of the Access Control Table.	56
3.2.4	Updating the Access Control Table.	58
3.2.4.1	Feasibility of On-Line Update.	58
3.2.4.2	Controlling the On-Line Update Process	59
3.3	THE SECURITY CONTROLLER MECHANISMS FOR ESTABLISHING A CONNECTION	60
3.3.1	Control Over the Initial Requestor-to-SC Connection. . . .	61
3.3.2	Selection of the Path for Set-Up Control Messages.	61
3.3.3	Handling Exceptional Conditions on Connection Creation . .	63

TABLE OF CONTENTS

Section		Page
3.3.4	Crossing Inter-Network Boundaries (Gateways)	64
3.3.5	The Contents of the Initial Control Messages	67
3.3.6	Control Over Play-Back of Connection Creation Messages	72
3.3.7	Implicit Connection Creation	72
3.3.8	Connections for Broadcast Messages	73
3.3.9	Connections for Unclassified Work	73
3.4	THE SC/HOST-LEVEL MECHANISMS FOR CONTROLLING CONNECTION USAGE	73
3.5	SC/HOST-LEVEL MECHANISMS FOR MONITORING	74
3.6	SECURITY ASSURANCE ASPECTS	76
3.6.1	Certification Issues	76
3.6.2	Handling of SC Data	77
3.6.3	Self-Checking	77
3.6.4	Physical and Procedural Controls	78
3.7	OTHER DESIGN ASPECTS	79
3.7.1	Network Control Programs	79
3.7.2	The SC Control Program	83
3.7.2.1	Basic Functional Modules of the SC	84
3.7.2.2	Auxiliary SC Functional Modules	85
3.7.2.3	Control Issues	86
3.7.2.4	Program Mechanization Issues	90
3.7.3	Error Control/Recovery in the SC	91
3.7.4	SC Hardware Requirements	97
3.7.4.1	Adequate Error Control Facilities	97
3.7.4.2	Security-Related Hardware Facilities	98
3.7.4.3	Operational Requirements	99
3.7.5	Performance Impact Due to Security	101
4.	NETWORK SECURITY AT THE ICD LEVEL	104
4.1	THE ICD IDENTIFICATION/AUTHENTICATION MECHANISMS	107
4.2	THE ICD ACCESS REQUEST/AUTHORIZATION MECHANISMS	108

TABLE OF CONTENTS

Section		Page
4.3	ACCESS CONTROL AT THE ICD LEVEL; ESTABLISHMENT OF CONNECTIONS.	109
4.3.1	Control Primitives	111
4.3.1.1	Insertion of Working Keys.	111
4.3.1.2	Handling Transparent Text.	112
4.3.1.3	Auxiliary Commands	113
4.3.2	Addressing of Embedded Control Commands.	113
4.3.3	Control Strings Using the Primitives	114
4.3.4	Concern for Errors in Control Commands	114
4.3.5	Notifying the Master ICD of the Connection Status.	115
4.4	ACCESS CONTROL AT THE ICD LEVEL; USAGE OF A CONNECTION . .	116
4.4.1	Encipherment Scheme Considerations	116
4.4.2	Crypto-Multiplexing Considerations	120
4.4.3	Control/Data Considerations.	121
4.4.4	Error Control.	123
4.4.5	Breaking a Connection.	125
4.4.6	Performance Impact Due to Security	126
4.5	SECURITY MONITORING BY THE ICD	128
4.5.1	Self-Monitoring of the ICD Operation	128
4.5.2	Checking for Improper Usage.	129
4.5.3	Augmenting the SC Monitoring Functions	129
4.6	SECURITY ASSURANCE ASPECTS OF THE ICD.	129
4.7	OTHER ICD ASPECTS.	130
4.7.1	Cost/Complexity Issues	130
4.7.2	ICD-Level Control Programs	131
5.	NETWORK SECURITY AT THE COMMUNICATIONS NET LEVEL	132
5.1	IDENTIFICATION/AUTHENTICATION ISSUES	134
5.2	ACCESS AUTHORIZATION CHECKING.	134
5.3	ACCESS CONTROL; ESTABLISHMENT OF A CONNECTION.	135
5.3.1	Initial Connection to the SC	135
5.3.2	Input Port Considerations at the SC.	135

TABLE OF CONTENTS

Section		Page
5.3.3	Bandwidth Requirements for SC Dialogs	136
5.3.4	Distributing the Working Keys	136
5.3.5	Notification of Connection Status	137
5.3.6	Ability to Establish Priority Connections	137
5.3.7	Establishing Connections in Process Addressed Nets.	137
5.4	USAGE OF A CONNECTION	138
5.4.1	Traffic Analysis.	138
5.4.2	Spoofability.	138
5.4.3	Denial of Service	139
5.4.4	Error and Flow Control.	139
5.5	SECURITY MONITORING	140
5.6	SECURITY ASSURANCE.	140
5.7	MISCELLANEOUS ASPECTS	141
5.7.1	Line Control Considerations	141
5.7.2	Network Terminal Handling Considerations.	144
5.7.3	Security Aspects of Different Network Architectures	145
5.7.3.1	A Dedicated Point-to-Point Net.	145
5.7.3.2	Circuit-Switched Network.	147
5.7.3.3	Tree-Structure Nets (Message-Switched).	148
5.7.3.4	Star Nets (Message-Switched).	148
5.7.3.5	Multiple Connected Message-Switched Nets.	149
5.7.3.6	Loop (Ring) Networks.	149
5.7.3.7	Radio Broadcast Nets.	150
6.	SUMMARY OF RESULTS.	152
BIBLIOGRAPHY.		154

LIST OF FIGURES

Figure		Page
1-1	A Simplified View of a Secure Network.	3
2-1	The Access Authorization Matrix.	17
3-1	The Levels Involved in a Secure Network.	41
3-2	An Example Requestor Profile	54
3-3	A Comparison of Linkage Addressing Schemes	65
3-4	Alternative Methods of Establishing the Working Connection	68
3-5	Connection Creation When Two SC's Involved	70
3-6	Levels Involved in the Connection Creation Process	71
3-7	The Request Control Block (RCB) Format	88
3-8	Step-Wise Cross-Checking of Two Processors	94
3-9	Alternative Interconnections for Checking of Duplicated Processors	96
4-1	An Exclusive-OR Encipherment/Decipherment Scheme	105
4-2	Simplified Conceptual Model of an Intelligent Cryptographic Device	106
4-3	Variations of Self-Synchronizing Schemes Using Cipher Text Feedback	118
4-4	A Pair of Encipherment/Decipherment Devices.	119
4-5	A Simplified Cryptographic Device and Its Equipment.	122
4-6	Two-Level Error Checking	124
5-1	Comparison of Character and Bit-Oriented Line Disciplines.	143
5-2	Difficulty Due to Crossing of Levels for ICD Usage with a TIP.	146

TABLES

5-1	Comparison of Network Architectures.	133

DESIGN ALTERNATIVES FOR COMPUTER NETWORK SECURITY

Gerald D. Cole
System Development Corporation
2500 Colorado Avenue
Santa Monica, CA 90406

ABSTRACT

The security problems associated with a network of computers are an extension of those of stand-alone computer systems, but require additional security controls due to the distributed and autonomous nature of the network components. The purpose of this investigation was to generate a pre-development specification for such security mechanisms by determining the issues and tradeoffs related to network security over a broad range of network applications, topologies and communications technologies.

The approach which was taken was that of utilizing a dedicated network Security Controller (minicomputer) for checking the authentication of requestors, and, to some extent, for authorization checking as well. The enforcement of the Security Controller functions would be by means of Intelligent Cryptographic Devices, which could be remotely keyed by the Security Controller when a requested communication was authorized. The Intelligent Cryptographic Device would incorporate the National Bureau of Standards Data Encryption Standard algorithm.

The investigation showed that this approach is a viable solution to the network security problems of a large class of computer networks, and that such security mechanisms should be developed for operational usage.

Key words: Access control; authentication; communication; computer networks; cryptography; encryption; security.

1. INTRODUCTION

In recent years, computer usage has grown to the point that it influences almost every aspect of our commercial and military environments. Concurrent with this growth has been the need to share resources, to better utilize expensive equipment, to utilize and build upon the work of others, and to share work efforts. This need for controlled sharing has grown not only in terms of the number of people involved, but also in the geographic dispersion of these people and their need for rapid access to and interchange of information. Such growth has presented new technological and operational problems in many areas, particularly in system security.

The first generation usage of computers created security problems which could be solved by using conventional physical, procedural, and personnel control methods. Sharing was basically a matter of dividing the computer usage into dedicated time-slots, with carefully controlled set up/tear down between jobs (or batches of jobs of the same security level). The development of multi-programming methods provided a more efficient mode of hardware use by rapid context switching between jobs and by overlapping operations. This development required the machine to execute several jobs concurrently, thus adding a new dimension to the security problem due to the multi-user environment.

As software and data base resources began to grow in size and value, the need to share these resources also became evident, and added another dimension to the security control problem; namely that of controlling access to the multi-resources. The next logical step in this evolution was to share such resources across two or more machines (systems), which introduced yet another dimension to the security problem. These multi-system networks present a solution to the problems of sharing which involve a large number of persons who are geographically scattered, but who require rapid access and interchange of information. Such networks present formidable security problems due to the multi-user, multi-resource, multi-system environment.

The purpose of our investigation is to define the security issues related to this complex network environment, and to determine the tradeoffs related to possible approaches and mechanisms which could resolve these issues. The end result of the study, as reflected in this report, is to be a pre-development specification with the scope as defined in the statement of work:

> "Analyze several computer network configurations with respect to their ability to support end-to-end security (protection of information from originator to final destination) on all possible communication paths in the network. The effort shall yield specifications which include communication protocol, switching techniques, and protection techniques at such a level that a secure network development may be specified and initiated."

An excellent starting point for the investigation was available in the paper by D. K. Branstad, "Security Aspects of Computer Networks" (BRA-73)*, which discussed many of the relevant issues. However, our study extended his efforts both in depth and by including a broader scope of issues.

Figure 1-1 is the general configuration of a secure network assumed for the purpose of this study. A set of computer systems (HOST computers), and terminals are to be interconnected via an arbitrary communications network, but under the control of a local "Security Controller and cryptographic devices

Assuming that each of the individual HOST systems is secure when operated in its own separate environment, we investigate the set of problems that occur when they are integrated into a loose federation, with certain global constraints and controls being placed over these otherwise autonomous local centers.

---

*All bibliographic references will be made by this abbreviated form of author and year of publication.

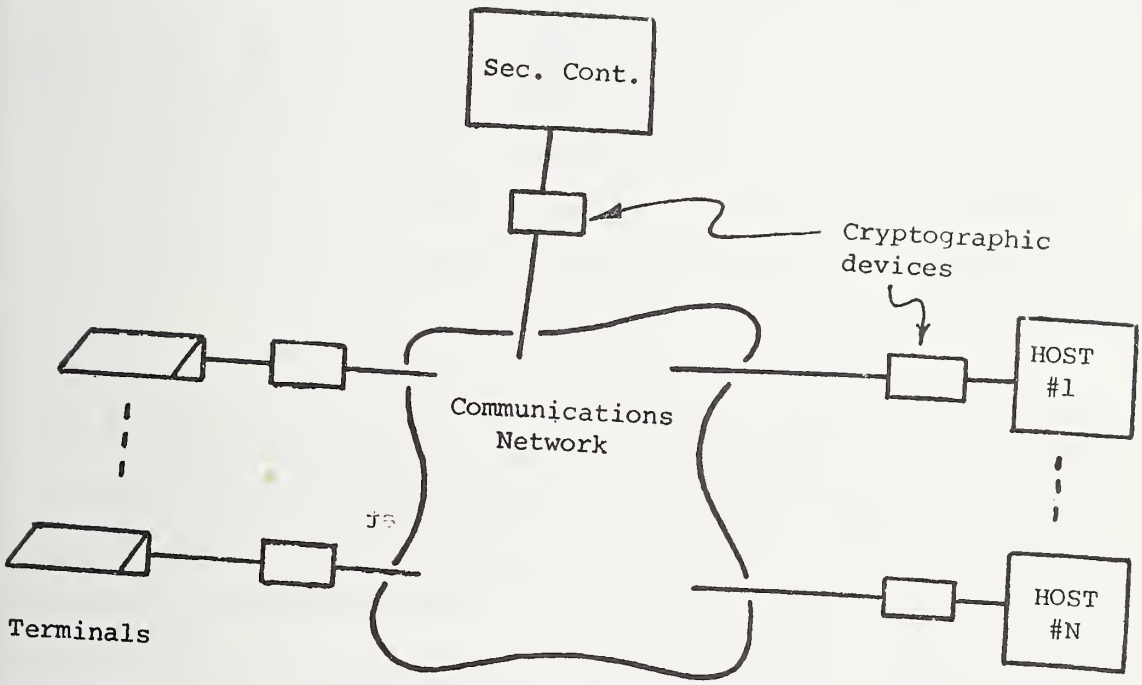


Figure 1-1. A Simplified View of a Secure Network

The need for cryptographic devices is apparent since the communications net is "open" to any would-be penetrators, but the cryptographic devices can provide much more than communications line security.

Unfortunately, many people feel that the use of cryptographic equipment "solves the security problem," while in reality this equipment should be viewed as only one element of a larger total system design for security. An Air Force Security Study panel summarized this matter as (AND-72):

> "Though considerable financial resources and management attention are drawn to the communications security aspects of networking (an important but well understood technology) the security problem of computer networking is not a communications problem but another more sophisticated instance of multi-level computer operating system security.
>
> "Currently, most secure computer systems achieve their security integrity by prohibition of multi-level and multi-compartment security operation. The computer is operated at a single, appropriately high security level for its needs, with all personnel and operating procedures controlled within the USAF/DoD established security framework. Networking ties two or more of these computer systems together; more often than not, systems dissimilar in equipment, configuration, purpose, management, and security control procedures. An example of the networking problem is the connection of the SAC SATIN network with AUTODIN network for both the receipt and transmission of information. Conceptually, the network can be viewed as a "supra-computer system." The network security requirements then are different than most of its members because the "supracomputer" operates essentially as a multi-level, multi-compartment, multi-user computer system. The network's security vulnerability is that each network node (i.e., the computer system operated by a participating agency) is unprepared

for multi-level, multi-compartment use by users over which it exerts limited, if any, control. Furthermore, the problem often goes unrecognized since management erroneously assumes security integrity because the supracomputer interconnections are via secured (often crypto) communications lines."

As mentioned in the preceding quote, the real problem area in network security is lack of global control over users, where "users" must be interpreted as any combination of persons and systems operating on their behalf. This loss of control can be reflected in any of several security problems, including:

1. Problems due to the large number of possible combinations of persons operating from different terminal stations at different sites with different authorizations for different resources at different sites, each of which has different classifications and compartments, etc.

2. N-th party problems in which processes operate on the behalf of a requestor, perhaps many levels removed, and may spawn other sub-processes, etc. (perhaps on several different HOST's).

3. The autonomous nature of each network participant creates problems in that each domain of control may have differing methods, interpretations, etc. for providing security.

4. The problem that one compromised HOST may be used to penetrate yet another (the "domino-effect").

5. The problems in which one operating system (or data base) may become faulty in a manner that spreads to other network HOST's (the network cancer problem; either accidental or malicious).

6. The potential problems related to a distributed attack on one system by two or more other systems (analogous to the asynchronous attacks on conventional multi-user systems).

These examples are by no means all of the network security problems, but represent the larger scope of the problems above and beyond that of individual resource-sharing systems. The examples also indicate that mechanisms are needed to prevent, or at least constrain, the spread of security compromises within the net. If this is not done, the network may not be any stronger than its weakest node, an unacceptable condition in any federation of entities. Network security must, therefore, be as independent as possible of the security of the separate autonomous nodes.

At this time, it seems appropriate to define what we mean by "computer network security." First, a computer network can be defined as an interconnected set of independent (or dependent) computer systems which communicate to share information and service resources in order to provide needed user services.* Dependence among computer systems may come about in any of several degrees, e.g., either directly dependent processes such as in a distributed computer system (FAR-73) or more subtly when computer centers begin to become increasingly dependent upon each other for services that would normally have been provided locally. Another area of emphasis is the definition on meeting user needs; A mechanism is of questionable value unless it meets the needs of its ultimate users, and this is one of the fundamental concerns in the investigation and specification of a secure network.

The definition of security in the sense of a secure computer network involves three basic aspects of protection: (1) providing controlled access to resources, (2) providing controlled use of those resources, and (3) providing assurance that the desired level of protection is maintained. At

*This definition was extended from that of Peterson and Viet (PET-71) and Farber (FAR-72).

this point we first encounter a question in defining the boundaries for "network security," which also reflects on our definition of a network--are the HOST computers considered part of the "network"? In theory, one should answer this question with an unequivocal "yes" (e.g., the supracomputer notion of the quoted ESD report); but in practice, one must often segment the problem into data processing and data communication aspects due to the autonomous nature of the local computer centers, and/or the administrative separation of data processing and communication areas. This investigation of computer network security is based on the following:

1. Both the data processing and communications functions will be considered as generally as possible.

2. The investigation will focus on the interface between the data processing and communications functions; i.e., the intermediate layer of equipment required when separately secure HOST systems are to be interconnected via an "open" communications network.

3. The investigation will also consider the resulting impact on both the data processing and communications in order to provide this secure interconnection.

4. Globally-defined network security mechanisms should augment rather than replace local (individual HOST) mechanisms. Aside from the "political" reasons (due to the autonomous HOSTs), augmentation rather than replacement also provides an evolutionary approach to network integration and the development of centralized security mechanisms which can gradually assume more of the total security functions.

We are then faced with two questions: (1) what global security-related policies must be developed to ensure network security; and (2) by what global and/or local mechanisms can those policies be implemented? The basic policy issues can be derived directly from our three-part definition of security:

1. Provide controlled access to resources

   All requesters of network services must be identified and authenticated, and their access request must be checked to ensure that it is authorized prior to establishing a connection (logical or physical) between the requester and the resource.

2. Provide controlled usage of these resources

   Although this is primarily the responsibility of the HOST providing the resource, the network interface must provide whatever functions it can to augment the HOST protective measures.

3. Provide assurance that the desired level of protection is maintained

   Two related areas of networking policy relate to maintaining a desired level of security: (1) monitoring or surveillance of network usage, and (2) ensuring the adequacy and integrity of the security mechanisms.

These three basic policy issues will be discussed in detail in later sections of the report, as will various issues and tradeoffs related to mechanisms which can be used to implement policies. These considerations can be viewed in a top-down manner by first exploring the policy, administrative, and requirements issues, which then reflect downward into HOST-level mechanism issues. These, in turn, help to define the cryptographic device level issues, which then further define the issues related to the communications network. Each of the major levels forms a separate chapter of the report to clearly separate the functions and tradeoffs within the separate mechanisms.

For each of these levels, one must consider all of the categories from the definition of security:

1. Identification/authentication
2. Access request/authorization
3. Access control; establishment of the connection
4. Access control; usage of the connection
5. Security monitoring (surveillance)
6. Security assurance (integrity)

These general topics become the sub-chapter headings within each of the top-down layers.

This report will (1) define the critical issues and problems that relate to network security, (2) describe the various mechanisms which might implement the policy/solutions, and (3) discuss the tradeoffs which relate to these mechanisms at each of the various levels.

The organization of the report is such that it can be separated into reasonably independent discussions of the policy and requirements, the global security control mechanisms, the distributed cryptographic mechanisms, and the communications network. Alternately, the issues related to the individual topics of authentication, authorization, etc. can be separated by selecting the appropriate sub-chapters, e.g., sections 2.1, 3.1, 4.1, and 5.1 for authentication issues. However, it is recommended that the document be considered in its entirety since even the individual, separable aspects should be viewed within the scope of the overall networking problems.

2. NETWORK SECURITY POLICY AND REQUIREMENTS ISSUES

The network security problem, like all security problems, exists because hostile elements would "misuse" certain valuable resources if given an opportunity. The nature of these hostile elements and the resources to be protected, leads to the development of appropriate policy issues and system requirements which, when implemented by security mechanisms, lead to some prescribed level of protection. This protection can never be absolute, and does not necessarily apply beyond some predefined set of threats.

We assume for this investigation that the nature of the hostile elements and the resources to be protected in a DOD environment is well known, and do not address these matters any further. However, the policy and requirements issues related to how these threats are to be countered are very much of concern since these matters establish the top-level constraints and requirements for our investigation, and thereby define what functions our security mechanisms must provide. Subsequent sections of this report will address the tradeoffs related to how these mechanisms might operate, but for the present, we will discuss what general forms of protection must be provided.

Many network security issues are straightforward extensions to those of any multi-user, resource-sharing computer, while others are unique to the multi-system environment of a network. These unique problems are the primary concern of this investigation, but in the interest of completeness, we shall also briefly describe the general problems. Similarly, other issues must be addressed if a secure network (or any network) is to be viable; for example, the concern for the user-to-network interface. These non-security issues will be discussed for areas which have been problems for other networks.

Certain matters under discussion must remain as generally open-issues*, since adequate solutions have not yet been defined (particularly in the areas related to heterogeneous systems). Where possible, we will recommend some action to close these issues, at least within a particular network environment.

2.1 IDENTIFICATION/AUTHENTICATION ISSUES

If a network is to provide controlled access of requestors to resources, the control mechanisms associated with these resources must have some way of determining and verifying the identity of the requestors. We use the term identification to mean the process of determining who or what an entity claims to be, and refer to the process of verifying this claim as authenticating (e.g., by using a password). The security aspects of concern are primarily those of authentication, since identification problems tend to be based on operational issues (e.g., whether Social Security numbers should be used as identifiers). Therefore, we will concentrate on authentication, addressing the following topics:

- Authentication of persons and devices
- Process and HOST level authentication
- Distributed versus centralized authentication checking
- N-th party authentication

2.1.1 Authentication Of Persons And Devices

All entities which can affect security must be uniquely identified and authenticated. In the most straightforward case, an entity would have a globally unique name and an appropriate authenticator. More complex situations arise for "composite entities" and environment-dependent entities. An example of the former is the attachment of an authentication device to a terminal. If such a device is non-forgeable, non-removable, and is otherwise adequately protected with physical and procedural controls,

---

*As a prime example of the open-endedness of many critical issues, consider the attributes of "security" itself, e.g., how can one quantitatively express the adequacy of a given approach to security.

it can provide an implicit authentication of the terminal. (The reader might argue that such a device thereby becomes part of the terminal. The issue is less clear when this authentication is provided by a cryptographic device, which will be the case in a later section of the report.)

We assume that environment-dependent entities such as a terminal which must be operated within a special room, are dependent upon physical and procedural controls to ensure that these restrictions are maintained. Alternate authentication-like mechanisms could be included (e.g., the terminal is in the room and the door is locked) by an extension to the notion of the attached authentication device as mentioned above.

If an entity is to serve a dual or multiple role, the two or more identification/authentication mechanisms required must be provided by some usage-dependent feature such as a special key, coded card, etc. For all practical purposes, such an entity would be viewed by the network as being separate, but mutually exclusive, devices. Here again, physical and procedural control of terminal access would be required.

In all of the above instances, the authentication must be made initially, on an on-going basis, and at any system discontinuities. The forms of authentication are varied, and depend upon the entity and the particular needs of the system. Reference (FAR-72A) gives a summary of existing identification/authentication methods.

2.1.2 Process and Host Level Authentication

Networking also creates identification/authentication problems beyond those of a single computer system. In the multi-system (network) environment, the various systems (HOST computers) must also be identified and authenticated. One aspect of this issue is whether processes on the HOSTs should be considered as entities which require such identification/authentication, either as a requestor of network services and/or as a server. Since the HOST

will have complete access to the data of its processes, including any authenticators which they might have, the use of process level authenticators does not protect against malicious HOST behavior. However, it does provide a degree of protection against accidental spillage (e.g., address error).

2.1.3 Distributed Versus Centralized Authentication Checking

Authenticators may tend to become less effective (e.g., more easily stolen and forged) in a network environment since passwords, etc. are needed at multiple computer sites and therefore tend to be: (1) the same at all sites, (2) of longer duration between changes due to the logistics problems in changing them, and (3) vulnerable to compromise at any one of the multiple sites (the "weakest link in the chain" effect). These problems are caused by the inherent weaknesses in distributed authentication checking where the authentication (e.g., passwords) can be forged if known. The solution requires either centralized checking or non-forgeable mechanisms.

The use of a centralized authentication checker (part of the "Security Controller" which we will define later) is in reality a hybrid scheme, with checking being distributed to the level of a given region or domain, but being centralized within each domain. Local checking is needed for logistics reasons, at both the user (requestor) and server (resource). However, this does not imply that only the centralized check would be made; distributed authentication checks could also be made, either as a two step check for particularly sensitive resources, or as part of an evolutionary approach to developing a secure network from a set of independent sites.

Such checking could also be by means of a duplicated facility in order to provide a secondary or back-up capability such that a failure in the primary checking mechanism does not result in a loss of network access. However, the logistics problems of managing a duplicated data base of users, passwords, etc. must be carefully considered. Some indication should also be made that network access is being made via the secondary source of authorization approval.

2.1.4 N-th Party Authentication

An additional aspect of authentication important in the network environment is N-th party authentication, e.g., when one site must operate on behalf of another, which itself is operating on behalf of yet another, but in all cases for some "ancestoral" user who initiated the request.

Two basic issues arise from this problem: (1) the extent to which the original requestor should be involved, and (2) the amount of information that should be carried along with the N-th party request. Addressing the first issue, the original requestor might:

- Explicitly specify the N parties involved
- Specify that some level of N-th party access is probably required, but with the parties left undefined.
- Not be aware of the need.

The first option is not generally realistic, and would typically apply only for certain single level indirect accesses. The second and third options are more likely, and require several aspects of protection including:

1. A method to notify a user that accesses are being made (or attempted) on his behalf. Even if accesses are transparent to the user, the fact that they are being performed may need to be made available, at least as an option.

2. A mechanism to ensure that the user can control accesses on his behalf, e.g., a one-time password, (given to the user by the SC), that the HOST would have to get (from the user) before being able to make an access on his behalf.

3. The same protection as in (2), but applied at each step of an N-th party scheme. (Should the user be involved in each step, or only at the first?)

4. Determination of the default conditions for N-th party accesses; i.e., whether allowed or precluded.

5. Some maximum number of levels for the N-th party accesses, e.g., N no greater than two or three.

The second basic aspect of N-th party authentication is how much information must be carried along at each stage of the chained requests. At a minimum, each stage must know the previous stage. The only other alternative which seems to have merit is that of carrying along information on all previous stages, i.e., a "trail" to the N-parties and their sequence. The advantages of this scheme are:

- To simplify audit data interpretation
- To provide an explicit "return path" for results
- To detect and avoid loops, (e.g., A calls B, who calls C, who calls A, etc.)

The "trail" alternative has the disadvantages of extra overhead and the open-ended length of the related data structure and storage requirements. Other aspects of the N-th party problem will be considered under authorization issues in section 2.2.4.

2.2 ACCESS REQUEST/AUTHORIZATION ISSUES

Identification and authentication typically precede the request for access to some network resource, since knowledge of the requestor is necessary to determine if the requested access is authorized. The information which defines the rights of requestors to access various protected objects (HOST computers, files, etc.) is basically the information indicated in a

Lampson/Denning (LAM-69) three-dimensional authorization matrix as in Figure 2-1, although actual mechanizations vary considerably from system to system. For our purposes, we will assume that every requestor has a capability profile (e.g., a "C-List") which consists of essentially the list of objects to which he has access, and the relevant privileges on those objects. Similarly, an object might have an access requirements profile (e.g., the list of requestors who have access to it and their privileges), so that an access request is authorized when the requestor's and object's profiles match.

The access profile information can be considered part of a more global security profile, which would also contain identifiers, etc. The term "profile" is used quite loosely in the literature, so we will not give it a rigid definition in this report. However, we will define profile-related information at various steps to discuss its possible form, content, size, error control, etc. as these relate to other issues.

Since the issues of access authorization can often be considered as extensions of those of authentication, we must consider many of the same general topics such as composite entities, N-th party situations, etc. These issues will be explored in the following sequence.

1. Access authorization design principles
2. Authorization checking at local and remote nodes
3. Component authorizations
4. N-th party authorizations
5. Checking required to access the authorization mechanism.

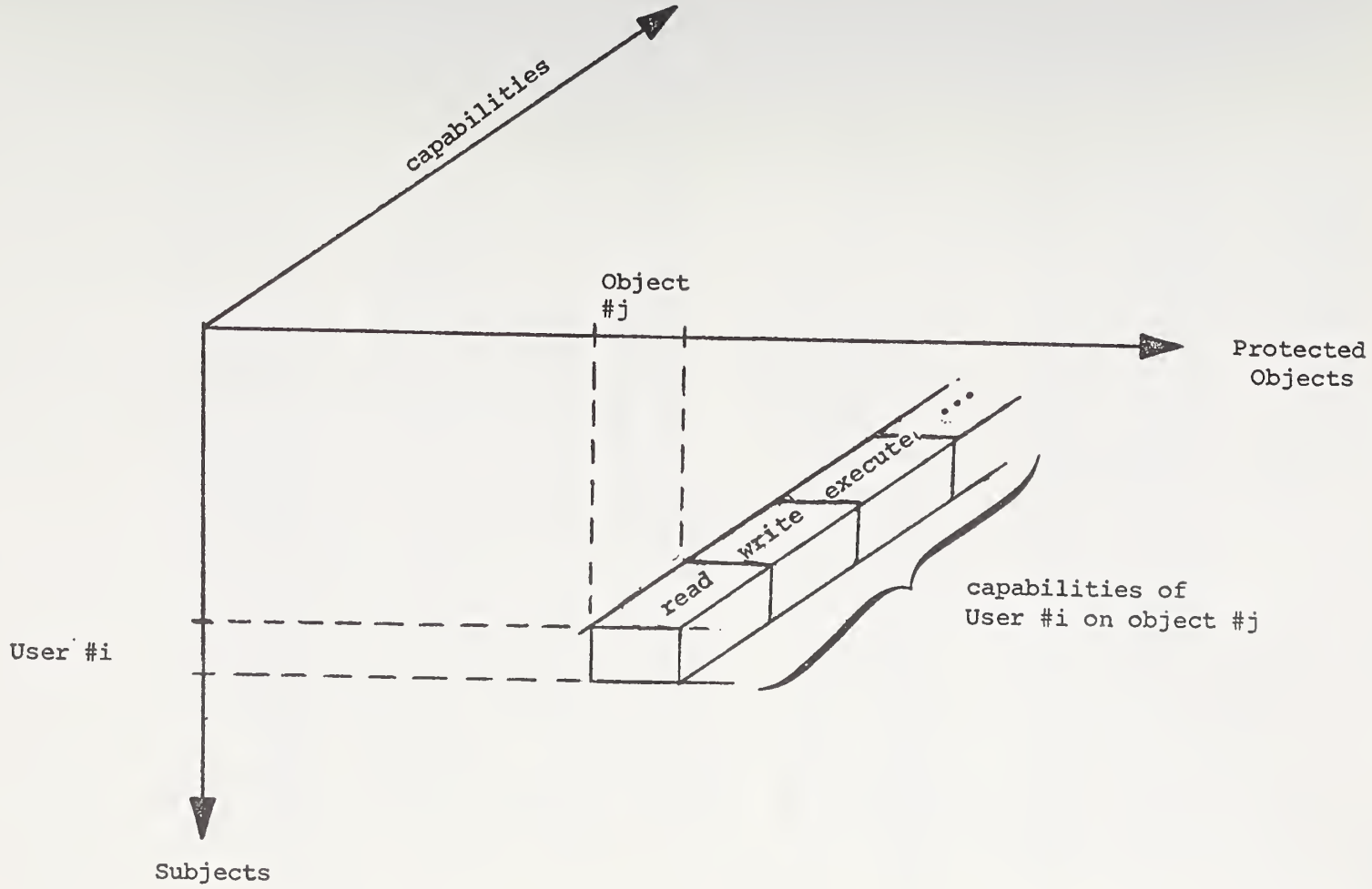


Figure 2-1. The Access Authorization Matrix

2.2.1 Access Authorization Design Principles

Authorization and access control are widely studied problems, and therefore will not be stressed here except to note the following design principles:

1. Least Privilege

   No requestor shall have any access privileges which are not required to perform its function (need-to-know). As a corollary to this, access to resources shall be separated (compartmentalized) whenever such separation adds to security (reference Jones; JON-73).

2. Least Common Mechanism

   There shall be minimal shared (common) mechanisms, other than those that are expressly there for security purposes (reference Popek; POP-74).

3. Reference Monitor Approach

   Access control mechanisms must be such that they are: (1) always invoked, (2) isolated from unauthorized alteration, and (3) accredited as being trustworthy. (Note: The Security Controller approach which we will develop is analogous to the Reference Monitor, but is only involved in the initial phase of the use of a resource.) (Reference Anderson; AND-72.)

4. Object Versus Path Protection

   Protection can be provided to either the object itself and/or the path to the object. (Note: The network aspects are almost entirely path-oriented protection).

2.2.2 Authorization Checking At Local And Remote Nodes

When a requested resource is in the same local domain as the requestor, the access authorization check can be made at that Security Controller by comparing the requestor's capabilities profile against the requirements profile of the resource. If the resource is not at the local domain, two additional operations are required: (1) discovering where the resource is, and (2) sending either the requestor's profile to the resource or vice-versa.

The first operation, finding the resource, can be resolved by any of the following methods:

1. Location explicitly provided by the requestor
   - Default condition of local node (analogous to a telephone area code)
   - Requestor provides name of HOST for a remote resource

2. Found via table look-up (directories)
   - Using local tables
   - Using tables in some predefined network service facility.

In the short term, network users could be expected to know the physical location of the requested resources, but provisions should be made for an evolutionary trend towards more implicit schemes such as the directory approaches. However, such service functions should not be mixed with critical security functions in a manner that complicates the security mechanisms and makes certification more difficult.

In addition to discovering where a resource is located, we are also concerned with the choice of which profile should be sent to the other's Security Controller for checking. The alternatives are:

- Send requestor's profile to the resource SC for checking.
- Send resource's profile to the requestor SC for checking.
- Do both, i.e., checking at both SC's.

The first option has a certain intuitive appeal--you take the key to the lock, not vice-versa. The second has a disturbing aspect since the requestor node checks its own request instead of having the protection (checking) at the remote node, which is presumably responsible for the resource.

Checking at both requestor and resource SC's introduces an additional level of checking, i.e., to see if both agreed that the request was authorized. This could be performed by sending both check-results to either SC (in which case the previous arguments apply again in a recursive manner), or by an implicit scheme in which each SC takes some separate action, which must match if the requested connection is to be usable. Since this added complexity has no apparent benefit, the scheme will not be considered further, and only the first method (checks at the resource SC) will be considered in subsequent sections.

2.2.3 Composite Authorizations

Several entities are involved in almost every computer transaction, e.g., a person, a terminal, a HOST computer, and a process. Each of these entities must be authorized to either receive, process, or transport the information being handled. The logical intersection of these authorizations* will

*In some situations, the authorizations may be other than the logical intersection, e.g., the use of statistical programs as discussed in the following section.

establish the level of information which can be sent via this sequence of entities (WEI-69), but a further step-by-step authorization check is also necessary to ensure that only the proper entity (or entities) are the ultimate recipients of the information, e.g., one entity may be authorized to process, but not to copy the information.

2.2.4 N-th Party Authorization

In some instances, a requestor will be connected to a HOST which will, in turn, need to access other resources on the requestor's behalf. This need can iteratively grow to the general N-th party authorization problem, which extends the previously discussed N-th party authentication problems. Authorization is a larger problem than authentication since the latter is strictly binary at each intermediate requestor. In contrast, the authorizations of each intermediate requestor may differ, as may the authorization needs when information is processed at the different nodes along the chain. Two different approaches are possible: (1) continually subsetting the authorizations as necessary so that the final privileges are the intersection of those of the original requestor and all intermediate nodes, thereby ensuring that no intermediate node gets any information for which it is not authorized (WEI-69), and (2) handling the authorizations iteratively on a pairwise basis, so that the N-th level will provide any requested information for which the N-1'st is authorized, and leave the burden of further controls on passing of data to that HOST. This approach allows the use of so-called "statistical programs" in which specific details are lost, e.g., "what is the average value of the class of xxx's," instead of "what is the value of a particular xxx" which might be sensitive. Of course, the latter may be the result of a cleverly devised statistical request, a well known problem that is also outside the scope of this network investigation. We consider the possibility of such programs since we want the network design to be such that it can accommodate new advances if and when they become available.

2.2.5 Checking Required to Access the Authorization Mechanism

There are two basically different approaches for access to the authorization mechanism: (1) access to use it, e.g., by any requestor, and (2) access to modify it, e.g., by special personnel who have such privileges. The former is considered the normal mode of operation since one of our requirements from the Reference Monitor analogy (section 2.1) was that the access mechanism be always invoked.

The second approach has particularly serious implications since another requirement was that the mechanism be isolated from unauthorized alteration. Three different schemes are possible:

- Allow access only by manual means (e.g., by authorized personnel acting under tightly controlled procedures).
- Use the authorization mechanism to protect itself, e.g., only certain requestors have the capability to change the authorization data.
- A combination of these two.

While the manual approach has an intuitive sense of control, its real viability will depend on the frequency and complexity of the updates, which could "over-whelm" the manual methods.

The second alternative distributes the control of authorization over a wider community of responsible agents, each of whom has some limited capability to modify the privileges of a particular group such as a project team. These agents might in turn be controlled by either some higher-level set of responsible agents,* or control at this level could revert to manual methods, e.g., in the combination approach.

Both schemes seem to have merit, and fortunately are not mutually exclusive. Therefore, the recommended approach is to develop the system with both features; using the third approach, i.e., manual controls over a set of agents (initially all changes), with the possibility of allowing these agents to delegate capabilities in some future version of the system.

*These levels of authorization define a hierarchy of increasing privileges, which is not universally accepted as being desirable, (e.g., see Wulf, et al (WUL-73) who claim that "such structures are inherently wrong and are at the heart of society's concern with computer security.")

2.3 ACCESS CONTROL; ISSUES RELATED TO THE ESTABLISHMENT OF CONNECTIONS

The proposed network security mechanisms gain a substantial part of their strength by their ability to control the creation of communication paths between a requestor and a resource. Establishing such a path or connection involves several different levels, some of them being conceptual (levels of abstraction) and some of them being physical (implementation levels which build upon each other). This notion of levels, their definition, separation, and transparency, is a critical aspect of networking which is as important as the analogous levels of the top-down design of a large-scale software system. It is doubly important for our investigation which has an intended generality in terms of network usage, HOST computer systems, encryption devices, and communications network technology.

"Establishing a connection" means that: (1) a logical or physical path is created through the communications net (e.g., for message-switched or line-switched nets respectively), (2) the appropriate control disciplines are initiated, (3) the encryption devices are keyed, (4) the requestor/resource control programs are initialized, and (5) certain identification and authorization data are sent to the resource. This sequence establishes a user-to-process (or process-to-process) communications path which has end-to-end protection and a defined set of capabilities. Three general areas will be considered relative to the creation of connections:

- The implications of HOST acceptance of the connection.
- The profile information to be sent at the time of connection establishment.
- Error handling on initial connection requests.

2.3.1 Host Acceptance of Connections

The controlled establishment of communication paths or connections is, to some extent, prima facie evidence that the requestor is legitimate and authorized to access certain resources. However, the HOST containing these resources may, at its option, present a further set of checks, thereby providing:

- Multiple "barriers" through which a potential penetrator must pass.
- An evolutionary development in which HOSTs can gradually accept the existence of the independent access controller.

2.3.2 Profile Information to be Sent at Connection Establishment

The information which should be sent from the security control mechanism to the HOST at the time of the connection creation might be any of several alternatives, e.g., the requestor's entire profile or only that subset which is relevant for the requested access, or perhaps none at all. Considerations on this tradeoff include:

- The requestor may not know in advance all of the resources that will be required.
- N-th party accesses may be required on the requestor's behalf, i.e., using some other resource.
- The particular subset of profile information may be resource-dependent.
- Sending profile information at the time of connection establishments may complicate the protocols.

This is an issue which will remain open for the time being since its impact on the network security mechanisms must be more carefully considered.

2.3.3 Error Handling

Since the creation of a connection and the accompanying profile information assumes that the requestor is authorized, the creation mechanisms must be secure even in their failure modes (i.e., fail-secure). Failures must be a primary design consideration in all of these mechanisms.

2.4 ACCESS CONTROL; POLICY ISSUES AND REQUIREMENTS RELATED TO USE OF A CONNECTION

All communications via the network must be via paths which have been explicitly created by the security control mechanisms described in the previous section. Considering the nature, use, and control of these paths, the following issues arise:

- Should the security control mechanism be part of the communication path?
- What control mechanisms are available via the encryption devices?
- What degradation is caused by the need for security mechanisms?
- How should data and control information be separated?

Each of these topics will be considered in the following sections.

2.4.1 Security Control Mechanism as Part of the Communication Path

Each protected connection should be established for a particular requestor/resource dialog, and should use a "one-time" encryption key assigned by the security control mechanism. However, this mechanism should not be part

of the communications path for actual usage of the connection, for the following reasons:

- The least privilege and least common mechanism arguments of section 2.1.
- The fact that the security control mechanism would thereby become a central node in a star-configuration network, which is known to have serious vulnerabilities and performance degradation.

Use of these connections therefore becomes independent of the centralized security control mechanism.

2.4.2 Control Via Encryption Devices

A degree of usage control can be provided by the distributed mechanisms, namely the encryption devices, if they have sufficient "intelligence" built into them. Such features should include:

- Protection against spillage due to erroneous addressing information or routing (e.g., by having different encryption keys for each requestor-resource pair).
- The ability to accept a new key from the security control mechanism for use in each separate requestor/resource dialog.
- Protection against improper use of a connection (e.g., by having a check within the encryption device to ensure that the "tagged" security level of the message does not exceed that for which the connection was established).
- Ensuring that sensitive data never appears in the clear within the net (including at any message-switching processors).

2.4.3 Degradation Due to Security Mechanisms

The security mechanisms should not unduly impact the network in terms of:

- User inconveniences (delays, uncertainties, memorization, etc.)
- Performance degradation (responsiveness and throughput)
- Error recovery at all levels
- Hardware and software costs (design, development and maintenance)
- Operational costs (administration and management of updates, etc.
- Loss of local control of resources.

2.4.4 Separation of Data and Control

The need to separate data and control is a basic problem in data communications, and is even further compounded when encryption devices are used in such communication links. Before considering this latter complication, let us discuss the problems involved in the clear-text handling of data and control, after which we will extend these notions to include encryption.

If multiple paths are available for information transfer, one can divide the set into data and control paths, with the latter having predefined interpretations such as timing, status, error indicators, etc. However, in many systems we are constrained to transmit both data and control information over a single path, and therefore must make the control versus data distinction by other means. Three generic possibilities exist: (1) to divide the class of possible symbols into data and control subsets, (2) to add a flag-bit to each symbol to indicate if it is data or control, and (3) to use mode control characters which switch back and forth between data and control interpretations of given bit patterns (DAV-73).

The added complications of encryption are due to the need to pass clear-text control information through (or around) the device to provide addressing information, etc. that the communications net requires to deliver the message. Passing control information through the device can conceptually be done in any of several ways:

1. By "disabling" the encryption for the desired interval of time, i.e., the key string is such that it passes the clear text without any change. The problems with the scheme are due to the "trap-door" that it provides, since the disabling may also happen under accidental or malicious circumstances.

2. By pre-encrypting the control information such that it becomes clear-text after passing through the encryption device.* This scheme also suffers from weaknesses involved in the generation of the pre-encrypted information, and to a lesser extent, the possibility that the meaningful control bit patterns may randomly occur as patterns as part of the pseudo-random output from the encryption device.

The above methods of passing control information through the encryption device give too much capability to the data processing equipment, so we will consider methods of passing this information around the device. The possibilities include:

1. A direct data path over which one may send arbitrary control information. This scheme has potential problems of misuse due to the general nature of the data that can be sent on the path.

---

*Since encipherment and decipherment by the exclusive-or are the same operation.

2. A direct data path, but with a predefined set of legal data elements which may be sent.

3. An "indirect" data path in which one specifies the name of a pre-stored control string to be activated (the pre-storing can be by either manual operations or by the security controller mechanism as part of the connection creation).

4. An implicit scheme in which only one control string has been pre-stored for use on all connections, i.e., only valid for dedicated point-to-point transmissions. The Private Line Interface (BBN-73, 74) is typical of this use.

Of the above methods, only the second and third offer sufficient protection and flexibility for general network use. Rather than selecting between these two methods, one can selectively use the best features of each, e.g., by using (3) for control information that must change for each dialog, and (2) for information that has a fixed representation (error indications). The design approach should therefore use these two techniques, selecting between them depending on the static or dynamic nature of the data.

Specific areas of control which must be addressed include those of timing, status indications, key control commands, exceptional condition indicators, and the control signals required for multiplexing.

2.5 SECURITY MONITORING ISSUES

Security monitoring refers to collecting information for: (1) gathering audit trail information on requests for network access, both granted and denied, and (2) detecting and aborting improper network use whenever possible, as well as being able to assess possible compromise when discovered after-the-fact. The second function requires global interpretation and control, and a Network Security Center is suggested to serve this role.

2.5.1 Collection of Audit Information

Collecting appropriate audit information is, at best, an art such that the tools to be provided must be left as flexible and open ended as possible. This is particularly necessary in the network environment since the information relating to a given use of network resources may become badly fragmented across the entities involved, e.g., if two or more security control mechanisms are involved in setting up connection(s) between a requestor and a resource, with possible N-th party iterations through other resources.

2.5.2 A Network Security Center

One possible solution to the problem of information dispersal is to centralize at least part of security monitoring into a "Network Security Center." Such a center should not preclude local checking of inherently local use, but could support the global needs of correlating and interpreting audit information.

One or more Network Security Center(s) could be formed, with audit information being collected by the security control mechanisms and HOSTs and sent to these centers via the network. This operation could be used either for routine audit processing, or on a more selective basis in which it would

be invoked, (1) when certain behavior patterns have been detected that require monitoring, and (2) for aperiodic pseudo penetration tests. The need for and implications of a Network Security Center will be left as an open issue since it is outside the scope of this investigation.

2.6 SECURITY ASSURANCE ISSUES

For our purposes, security assurance can be defined as those aspects of a system design and operation related to the adequacy of the security mechanisms, and the confidence level which we have in the integrity of these mechanisms. We therefore have as objectives:

- Accreditation that the mechanisms provide a given set of protection capabilities.
- Determining the sufficiency of the protection provided.
- Determining the need for security of the device design.
- Ensuring adequate reliability such that the mechanisms are available to requestors, but having known failure modes that do not erroneously grant access.
- Providing self-checking to further ensure that the mechanisms are operating properly.
- Ensuring proper interface to physical and procedural controls.

These security objectives must be considered at each of the various levels of the system design (both abstract and physical), and at each epoch of the system use (e.g., system generation, initialization, restart, and shut down). Care must also be taken to ensure that each design decision is fully consistent with the fundamental security requirements, rather than merely automating the existing manual/paper-oriented methods which attempt to meet these requirements. These latter methods should be considered as potential analogies to how the system might operate, but not as requirements in themselves

2.6.1 Accreditation of the Security Mechanisms

Overall security can never be absolute, nor can the accreditation of any individual security mechanism be determined with complete certainty. Even with the use of "proof of correctness" techniques, we can never be completely assured that the proof itself is correct, or that an implementation necessarily matches the more abstract primitives of such a proof. Therefore, accreditation or certification of a system must be based on the best available design, development, and implementation techniques, coupled with thorough tests to demonstrate empirically that these techniques have been and continue to be effective. Both hardware and software must be subjected to an adequate re-accreditation after any changes.

2.6.2 Sufficiency of Protection

Any design must be based on a set of requirements, and can not necessarily be expected to meet any needs that have not been included in this statement of requirements. Similarly, a security mechanism can not be expected to protect against threats which were not considered in its design. Therefore, the completeness of the design is a crucial issue, and is one that requires a continual reassessment.

Security mechanisms must also be extendable if they are to handle new and changing requirements, which otherwise will lead to administrative circumvention of the mechanisms. The following are test cases to check the extendability of a given design approach:

- The mobile user who needs to move and still maintain access via the network.
- Users with "dual roles," e.g., with differing privileges and needs based on some context.
- Handling security domains which fall within the "domain of control" of two or more security control mechanisms.

2.6.3 Secrecy of the Mechanism Designs

A controversial issue is that regarding the sensitivity of security mechanisms, as opposed to that of their priming keys (or initialization values). The argument that such mechanisms should not be secret, but should be openly discussed, has been raised by a number of authors, including Baran (BAR-64) in writing about networks and Weissman (WEI-69) on the ADEPT time-sharing system. The controversy is primarily one as to whether any additional security is to be gained by keeping the operation of the mechanism secret, in addition to keeping the initialization values secret.

The arguments for keeping the mechanism secret are that this provides a second, and independent form of security. That is, the penetrator must not only discover the encryption key, but must also determine the encryption mechanism that is being used. This approach has two basic weaknesses: (1) it precludes the usage of a standard encryption algorithm (such as the NBS Data Encryption Standard), and (2) it is extremely difficult to maintain the secrecy of the mechanism due to the large number of persons involved in the design, development, usage, and maintenance of a device. As a consequence, the principal security strength must be in the key and its secrecy.

2.6.4 Reliability and Failure Modes

Two major types of errors are of concern: (1) granting access when it was unauthorized, and (2) failing to grant an authorized request. The latter is of lesser concern by one or more orders of magnitude, depending on the circumstances. Reliability impacts both of these types of failures; the first in the modes of failure when they do occur (the need for fail-secure operation), and the second in terms of the frequency and duration of failures (since during such intervals, authorized users are denied service or must be handled by some backup scheme).

The design must consider all modes of failure of both hardware and software and how such errors will be detected and corrected. Redundancy must be applied, particularly in those areas in which failures would grant unauthorized access.

2.6.5 Self-Checking

The proper operation of the security mechanisms should be verified on an on-going basis by means of both diagnostic and pseudo-penetration programs. These checks should be made with an appropriate frequency to detect and limit the extent of any possible compromise due to failures. This provides a second level of checking to backup the fail-secure intent of the design.

2.6.6 Interface to Physical and Procedural Controls

The security mechanisms must have a certain degree of protection in both controlling their use and their modifications (either updating tables of information or making corrective changes). These considerations must be taken into account in the specification of the security mechanisms.

2.7 OTHER POLICY AND REQUIREMENTS ISSUES

Several other aspects of the design and development of a secure network will be discussed in this section. These are often issues that involve non-security areas, but which must be considered if the network is to be viable. They include:

- The user interface to the network
- Network management
- Meeting network traffic needs
- Separation of data processing and data communications.

2.7.1 The User Interface to the Network

One of the most critical (but often overlooked) aspects of resource sharing, and particularly of resource sharing networks, is that of providing a viable user interface. While this issue is not within the scope of our investigation, it deserves mention here to ensure that it is kept as a fundamental network design consideration. Authors who have addressed this problem, and thereby provide a good starting point for additional efforts, include Hicken (HIC-70, 71) in writing about the COINS network, McKay and Karp (in RUS-72) on the IBM Computer Network/440, and Pouzin (POU-73) on the CYCLADES network.* The underlying problem is, as pointed out by Hicken, that of differences. Differences in languages, character sets, conventions, etc. are often left to the user to resolve, thereby creating a significant problem for the experienced user, and an almost hopeless situation for the more typical user who is not a computer expert. While the security control mechanisms should not attempt to solve the user-oriented problems of the HOSTs (e.g., by tutorial, directory, or other services), its design should consider the varying usage-experience, typing abilities, etc. of the user community, and provide sufficiently clear requests and commands in the security control/user dialog.

---

*Other user-oriented references are by Neumann, (NEU-73 B and C), Pyke (PYK-73) and a bibliography by Blanc, et al. (BLA-73), revised and reissued by Wood, et al. (WOO-76).

2.7.2 Network Management

A second major area of concern that affects network viability is that of network management. Even without the concerns of security, networking presents significant administrative, economic, and procedural issues such as:

1. Lessened local control, autonomy, and self-sufficiency.

2. Committee-oriented decisions on how policy is to be implemented (and perhaps in the development of the policy itself). Such decisions tend to be, at best, democratic and do not necessarily address the long-term needs.

3. Network accounting practices are complicated by a number of factors such as: (also see NEU-73)

   - If the resource to be used is selected explicitly by the user or implicitly based on loading, etc.

   - If the "unit of currency" is dollars or some comparable amount of service to be given in return

   - If non-uniform accounting procedures are used at the various sites

4. Network membership standards may need to be developed to ensure a certain base-level of services, availability, documentation, etc.

The need for security adds other requirements and complications such as:

- The need for uniform (or at least consistent) definitions of security levels, user ID, authentication mechanisms, etc.

- Concerns as to whether accounting information contains any sensitive use information (analogous to "traffic analysis").

2.7.3 Meeting Network Traffic Needs

The network must provide adequate facilities to meet the responsiveness and throughput requirements of its users. The diversity of these needs may or may not be a problem, depending on the manner in which resources are allocated (e.g., dedicated lines, switched lines, or dynamically multiplexed such as packet switched). These latter tradeoffs will be discussed in Section 5, but our concern at this point is to establish the types of traffic that must be handled.*

One such type of traffic is control. These messages will typically be very short, will require rapid response, and will be sent asynchronously with respect to regular information flow. A second type of traffic is what we might call interactive or conversational, and a third is bulk traffic such as file transfers or I/O streams. The needs of these three types of traffic vary, with the first needing very rapid set-up but relatively low data transmission bandwidth, while the third has just the opposite requirements. Conversational needs are at some mid-point between the other two, leaving us with the requirement for providing a broad spectrum of network capabilities

Actual use of the network is also assumed to be of a wide diversity of applications, including batch (remote job entry and output), interactive, and computer-to-computer resource sharing. The level of activity is assumed to be quite heavy, probably supporting a few thousand users. The intent of the specification and the design tradeoffs is to provide generality, either directly or via expandability along open-ended design features.

2.7.4 Separation of Data Processing and Data Communications

Few people would disagree with the notion that there should be a clean separation between data processing and data communications; the argument comes about when one tries to draw the line that separates the two. For

---

*Adapted from reference POU-73.

example, is a Front End Processor (FEP) part of the data communications or the data processing? Does this answer depend on the functions the FEP performs, e.g., off loading HOST* functions of buffering, preprocessing, character translation, etc.? What if these same functions were provided in an IMP-like* device; would the IMP then become part of data processing? Conversely, are the network control programs* in the HOST's part of the data communications? Are they, when they are part of the TIP (Terminal-IMP)? Does encryption equipment always fall on one side or the other (or perhaps define the separation)? The list could go on and on.

If the dividing line between data communications and data processing is so ill defined, why does it really matter? It matters to some people because their very jobs may depend upon it. It matters to others because they want to be involved in the problems of one area to the exclusion of the other. It matters to those who must integrate the two disciplines, and they are typically the only ones who see it as it really is--not one dividing line, but rather a series of layers in which each layer is built upon the one below it, with the lower levels becoming increasingly transparent to its design and operation. (Ideally, a level is dependent only upon the one level directly below it, but some lower level attributes must be considered in most real-world designs.)

If the notion of a network being viewed as one entity, a "supracomputer" (AND-72), is to be met, the functions of data processing and data communications must be considered as part of an integrated overall system design. Only with this level of control over the design and development can we reach a meaningfully viable and secure network.

---

*ARPA Net terminology.

3. NETWORK SECURITY MECHANISMS AT THE SECURITY CONTROLLER/HOST LEVEL

The previous chapter discussed general policy matters and requirements that must be considered in the design of a secure network. These issues form the top-level constraints, guidelines, measures of quality, etc. that will be needed in order to determine and evaluate the tradeoffs related to possible network security mechanisms. These mechanisms will be distributed across a number of entities, which occur at different levels of the network structure as shown in Figure 3-1.

In addition to the usual terminals, HOST computers, etc. that would be expected in a network, we have added a new HOST-level entity called the Security Controller (SC), which will mechanize those third-party control functions described in the earlier requirements section. Since our concern is primarily with network security issues beyond those of a single/HOST system, our major concern at the HOST/SC-level will be with the Security Controller. However, we will consider the HOST and terminal subsystems to the extent necessary to present an integrated view of the overall network.

The HOST systems are considered to be the composite of hardware, operating system, program, and data resources, and include a variety of mechanizations for sharing of such resources, e.g., processing RJE work, offering time-sharing services, etc. In addition, some of the HOST's could be considered to be "mini-HOST's" in the sense that they offer some minimal direct service to their users, but primarily provide terminal or RJE access to the network and its wide variety of resources.* Regular HOST's may or may not have directly attached terminals: but as shown in the figure such terminals can not have the same end-to-end protection as provided for a terminal with a dedicated Intelligent Cryptographic Device (ICD).

---

*The mini-HOST would differ from an "intelligent terminal" in that the mini-HOST would multiplex its services over a set of conventional terminals, while the latter is normally viewed as one composite entity. For our purposes, the primary concern is the effect of multiplexing, so the intelligent terminal would be treated like any conventional terminal.

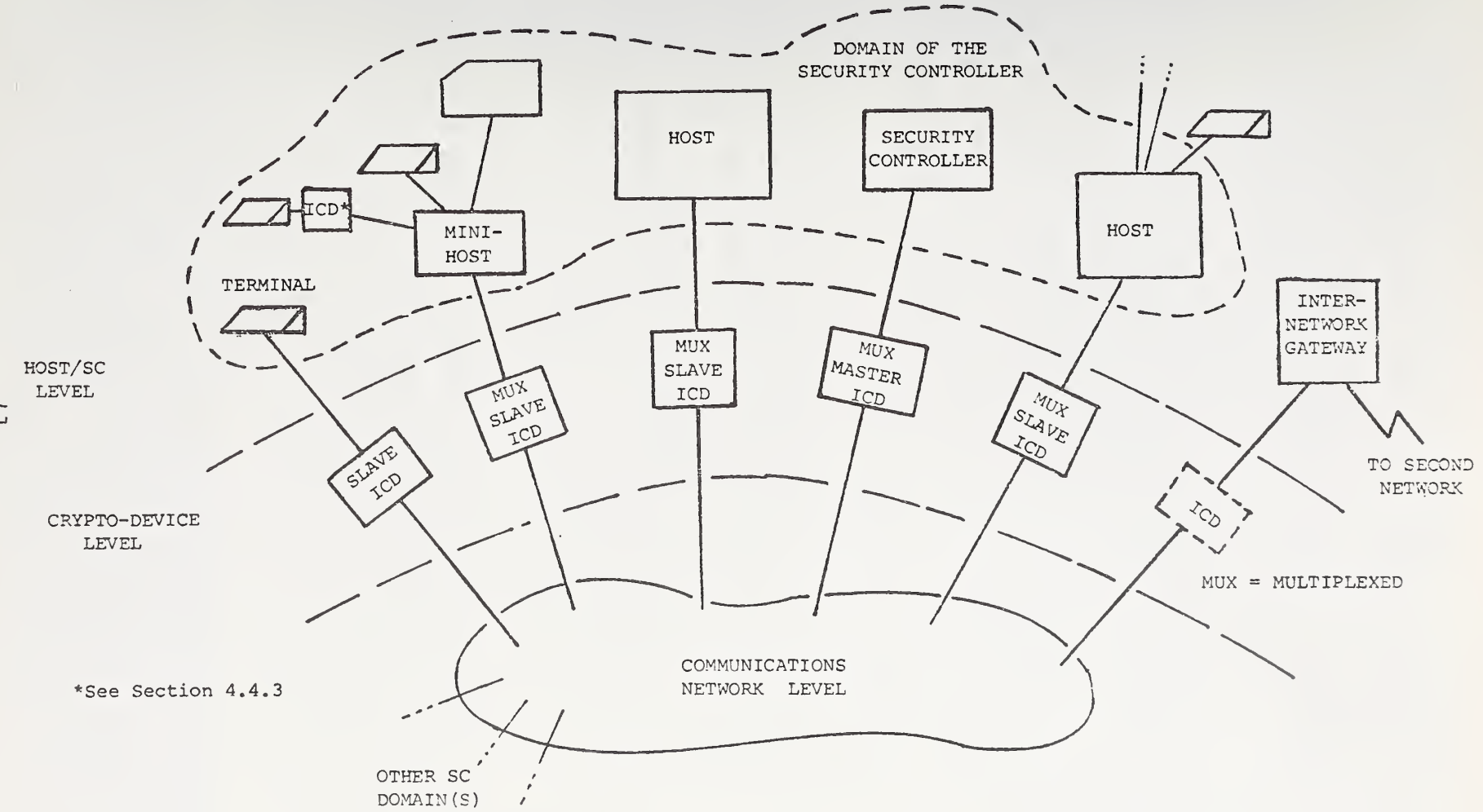


Figure 3-1. The Levels Involved In a Secure Network

The Security Controller is a refinement and renaming of what Branstad (BRA-73) referred to as an "Agency Computer." The latter name was based on the notion that the device would operate as the person's agent, but was changed to Security Controller to stress the fundamental purpose of the device, namely to control network security. The name, agency computer, might also imply that it would perform certain user-oriented services other than those of security control, (e.g., to provide directory services, tutorial aids, etc.) which are not consistent with the need to minimize its potential security flaws. In this spirit we will limit the functions of the Security Controller to those which are directly related to its role as an independent (third-party) mechanism, i.e., (1) to ensure that requestors are valid (authentication), (2) that they have legitimate access to resources (authorization), (3) that a secure communications path is established for their usage, and (4) that appropriate information is collected and/or distributed relative to this connection (audit data collected, user profile information to the HOST, etc.).

A given Security Controller will have some domain of control based on the resources and user population for which it is responsible. This domain could range from completely distributed (a separate SC per HOST system) to completely centralized (a single SC for the entire net). Intermediate solutions could include any arbitrary subdivision, e.g., based on network topology, geographic distribution, or administrative boundries.* Such subdivisions would form the individual domains for the various Security Controllers as indicated in Figure 3-1.

One other entity that would be needed if a network were to be interconnected with other dissimilar networks is also shown in Figure 3-1. This internetwork interface will be referred to as a "gateway" and would be needed due to differences between two nets in one or more protocol levels. These differences would be mediated by the gateway via transformations of the message

---

*For purposes of discussion, we will typically consider the subdivision to be on the basis of topological structure, i.e., one SC per local subnet, but there is no inherent need for physical subsetting. For example, an ARPA-like network could be arbitrarily divided into subsets of HOST's, with each subset forming the domain for one Security Controller. This flexibility is due to the way in which an entity is "attached" to an SC, namely by providing that SC (and only that SC) with the value of a private code (key) that is associated with that entity.

leaders, formats, etc. with the gateway appearing in an intermediate destination HOST (to limit the affected protocol levels to those of a HOST). If such transformations are feasible for the two nets, the gateway would effectively reflect a foreign HOST as a pseudo-HOST at the local subnet. The gateway will be considered further in Section 3.3.4.

The primary concern of this chapter will be with the specification of the Security Controller. Other entities such as users at terminals, HOST's, etc. will be considered as they affect the SC, but will not be our concern per se. The next lower level entities, the Intelligent Cryptographic Devices, will be mentioned as needed to define the interface and the requirements that are passed down by the SC, but will otherwise be deferred to Section 4.

3.1 IDENTIFICATION/AUTHENTICATION

The Security Controller will perform the necessary identification/authentication checking to determine who the requestor is and to ensure that the claim of identity is correct. This checking must be done for all requestors, including persons at terminals, HOST computer systems, and other SC's.

The form of authenticators will vary depending on the type of entity; with persons having memorized passwords, special badges (electrically readable), or unique personal characteristics which can be sensed electronically, while HOST computers may require quite different mechanisms for their authenticators. Due to this variety, and expected future developments in this area, authentication should be considered a separate and self-contained module of the SC which receives an input and produces GO/NOGO results, but is otherwise left completely open-ended. For our purposes, we assume that authenticators (at this level) are in the form of some arbitrary bit string such as a password.

3.1.1 Identification Information

Different entities require differing amounts and types of identification/authentication information. For a person, this information might include the following (which has been adapted from a list by Bushkin (BUS-74)).

- The person's name.
- A unique permanent identifier.
- The user's organizational affiliation.
- The user's organizational assignment.
- The user's citizenship.
- The computer system to which the user is primarily assigned.
- All information necessary to authenticate the user (e.g., passwords).

For a terminal, mini-HOST, HOST computer, or Security Controller, this information might include:

- The address of the device.
- A unique permanent identification number of the device.
- The physical location, including the building, room number, and nearest telephone number.
- The cognizant organization and the computer system to which it is primarily assigned.
- The person responsible for the device, his organization assignment, physical location, and telephone number.

The last three items would be primarily used by a security officer if audit information indicated some suspicious behavior at the device.

3.1.2 Providing Network-Wide Authentication

One of the critical networking problems is how to provide authentication of requestors across a number of server sites (e.g., HOST computers). The options are basically:

- Provide a common authenticator for all members of a given set of users, and have this authenticator stored at all appropriate servers.

- Provide a unique authenticator for each user, to be stored at all appropriate servers.
- Provide pair-wise unique authenticators, i.e., a different authenticator for each requestor, server pair.
- Provide an independent (third-party) authentication mechanism which is known and trusted by all servers and can itself be authenticated to them.

The normal solution is the second method, that of a single authenticator per requestor and with this authenticator being distributed to each potential server site. The pair-wise unique scheme is not viable for most users due to; (1) the large number of authenticators that would have to be memorized and (2) the occasional need to utilize resources in a manner that is transparent to the user (e.g., to select a server for a batch job based on load-sharing considerations).

The most viable and secure approach is that of utilizing a known and trusted third-party which itself can be authenticated. This third-party in our current approach is the Security Controller, a small computer that is dedicated to security functions.

The SC provides a centralized point for authentication within its domain and thereby eliminates (or at least augments) the weaker distributed scheme. After the SC has authenticated the validity of a requestor (and the requestor's access authorization as discussed in Section 3.2), the SC must send this information to the server site and must therefore be able to authenticate itself to the server. This could be done by a pair-wise unique password, i.e., that is known by the SC and the HOST. However, we can also utilize an implicit authentication scheme which takes advantage of the SC's involvement in the remote keying of the ICD's and the intimate relationship between the ICD and the HOST/SC. This latter aspect will be considered in Section 4.

3.1.3 The SC as a Pre-Connection Check or a Reference Check

The SC can serve its authentication role in either of two configurations: (1) as a pre-connection check in which the requestor must pass the SC's tests before becoming connected to the HOST, and (2) as a reference check for the HOST. In the second case, the requestor would initially connect to the HOST and present an authenticator (e.g., password), which the HOST would relay to the SC for checking. This approach would inherently include the handling of terminals that are hard-wired to the HOST, would allow a degraded mode of operation (dead SC), and might give some apparent increase in confidence that it was actually the SC that the HOST was talking to. However, the negative aspects of the "reference check" approach outweigh any advantages; namely (1) the requestor gains direct access to the resource that is to be protected, (2) the initial cryptographic keying functions of the SC are lost (they will be shown to add significant security), and (3) the HOST sees the requestor's authenticators that only the SC should know (least privilege). Therefore, we shall only consider the pre-connection mode of operation in further discussions.

3.1.4 SC-to-SC Authentication

When a network includes two or more Security Controllers, a method must be provided for inter-SC communications, which must also be established in a controlled manner. The four basic authentication strategies of Section 3.1.2 also apply here, and in the context of the Security Controller become:

- Use of a common code (key) known only by the SC's.
- Use of a unique code (key) for each SC, that is known by all of the other (server) SC's.
- Use of pair-wise unique codes (keys), i.e., each SC-to-SC pair has a unique value.
- Use of a Super-SC to mediate SC-to-SC connection requests.

The first scheme is particularly weak since its compromise would allow a penetrator to imitate SC-to-SC requests to any or all SC's. The second

scheme is only somewhat better, being basically the same as that utilized by many networks today in which a given requestor's password is known by all appropriate sites.

Use of pair-wise unique codes avoids the problem of the first two schemes, and is a viable approach for a relatively small number of entities, as would be expected in the case of the SC's. The fourth approach would apply the Security Controller notion recursively to create another level of control, a super-SC, that would control SC-to-SC connections. This scheme was suggested by Branstad (ibid) as a possible solution, and although it appears to be feasible, it seems to be an added complication that is not warranted by sufficient added security, and it also introduces a centralized mechanism which must be extremely reliable, since if it is down, it would cause a loss of all inter-domain network usage. Therefore, we have not pursued this hierarchical approach, but have instead assumed that SC-to-SC controls would be handled by the SC's themselves, with the pair-wise unique codes or keys being the most viable and secure solution of these four possibilities. Knowledge of either passwords and/or cryptographic keying parameters can be utilized for this purpose.

3.1.5 The SC Role in N-th Party Authentication

The general aspects of N-th party authentication were discussed in Section 2 and four requirements were derived which will be expanded here in terms of what the SC can do to help mediate this situation. These four requirements were:

- A method is needed to notify a user of attempted access on his behalf: The SC could be designed such that it would notify the original requestor of such N-th party accesses, although other constraints may not make this feasible to carry out. For example, a user's terminal may have a single "port" which is busy (i.e., tied to a HOST), or may not be able to receive calls (initiate only). Similarly, the process may be a deferred batch job in which case the user may not be on the system at the time.

Therefore, the only generally applicable solution seems to be to notify the requestor after the fact, e.g., as part of the audit information and printed with the eventual output to the user. Note: a subverted HOST could delete the print-out record, but could not nullify the SC's audit log.

- A mechanism to ensure that the user can control access on his behalf: Depending on the default condition chosen, the user controls would be either (a) allowed unless explicitly precluded, or (b) precluded unless explicitly allowed (or some combination).

- The same as the second requirement, but applied at each step: The same basic notions apply, although the number of possible combinations increases, e.g., usage of a set of one-time passwords or the same one repeatedly, the possibility of involving two or more SC's (if the N-th party servers are in different domains), etc.

- Some maximum number of levels of N-th party access: This choice would be very dependent upon other decisions, e.g., whether the SC was to set up and retain N one-time passwords for the subsequent N-th party accesses. (The approach is analogous to the manner in which some computers limit the number of levels of indirect addressing.)

The general problem of N-th party authentication is complicated by the combinatorics of the situation, and requires additional research to establish a proper conceptual framework from which adequate controls can be developed. However, there does not appear to be any conflict between the Security Controller approach and the known N-th party requirements.

3.2 ACCESS REQUEST/AUTHORIZATION

After the SC has identified and authenticated a requestor, it must determine if the requested access to resources is authorized. This check is basically a table look-up operation, and therefore raises the interrelated concerns of: (1) what information should be in the table, (2) how the table should be organized, (3) how the table should be utilized, and (4) how table updates should be made. We will consider each of these areas in the following paragraphs.

3.2.1 The Content of the Access Tables

Access authorization can be viewed as a three-dimensional access matrix of a set of subjects having access to a set of objects, and having a set of capabilities on those objects as described in Section 2. Complications begin to surface when one considers the actual meanings of these three axes. For example, a subject may be a person, or the composite of a person and a terminal, or perhaps of the person, terminal, and process(es) operating on his behalf. Other subjects may have no specific person involved (i.e., a process which performs a function but does not do so on behalf of a specific person). Also, some processes may operate with higher capabilities than the person on whose behalf they are operating (e.g., statistical programs that produce "open" results from sensitive data). These factors present complications and issues which must be addressed.

Let us first consider the subject axis of the matrix. Certainly persons must be included as entries since persons are the "ultimate consumers" of sensitive information, and are the only accountable entities in the network (at least the only ones that are punishable). Since the composite of a user and a terminal may have some lesser capabilities than that of the user (or conceptually some greater capabilities), then we must decide how this composite capability will be handled. Clearly, we do not want to consider each user-terminal combination as a subject; the list would rapidly become too long. What we really want is the composite capability, which can be obtained by a

combination of table look up processing. Therefore, the list of subjects should include: persons, terminals, and processes; and the composite capability of any combination of these subjects can then be computed based on the appropriate circumstances. Note that this structure will allow userless or terminal-less processes, userless terminals, etc., leaving to the table entries and computing algorithms the matter of enforcing a given access control policy.

The object axis will include any or all of the subjects, as well as HOST computers, files, etc., depending on the level of access control expected of the SC. Initially, only HOST-level access would probably be controlled by the SC (although access to certain files might also be handled) but the mechanisms should be sufficiently general to allow later extensions to other objects.

For each subject-object pair, a set of capabilities (possibly null) defines the privileges which that subject has to that object. Like the subjects and objects, each capability must have a unique global name, i.e., WRITE must have the same meaning for any subject - object pair. (Note: This requires that each network HOST computer be able to map these globally defined terms into its own access control interpretations if the SC is to be able to authorize accesses on a conditional basis such as "can append to a file, but not modify.")

3.2.2 Organization of the Table

The three-dimensional matrix is a conceptual model of the access control structure, in which the access privileges of a subject to an object are defined in the vector associated with that (subject-object) pair. The composite of all such capability vectors for a given user results in a plane of objects and capabilities, just as a particular object has an associated plane of subjects and capabilities.

While this three-dimensional structure is a useful conceptual model, it is not suitable for any kind of direct implementation. Even though one could map this structure into the single-dimension address space of a computer, the matrix is typically very sparse, and only the non-zero triples of user-

object, capabilities need be stored. Secondly, a reasonable amount of factoring is possible such as by listing the object-capability pairs for each user, or conversely, listing the subject-capability pairs for each object.

Other forms of compacting are possible such as identifying common access groups, where an access group is defined to have a specified set of object-capability pairs. Any subject having all of these pairs is considered to be a member of the group, and that subject's profile need only list these groups (by assigned group names), thereby reducing the amount of information that must be stored, at the cost of some additional processing time.

One standard scheme of grouping is by need-to-know categories (NTKC's). It is an implementation of the principle of least privilege that applies particularly well to manual/paper and pencil security environments in which persons can access (read) and perhaps generate sensitive information if they are authorized, i.e., a member of the NTKC. The concept of NTKC's becomes more complex in a computer environment in which many different privileges (capabilities) may exist, the files of the NTKC may reside at different HOST computers, etc. However, a sufficient generality can be obtained by considering object names to be of the form: <object name> = <NTKC>·<file>, in which various fields can be null, i.e., implied factoring of all such entities. Therefore, an object can be a NTKC, a HOST computer, a file within a given HOST, etc. Various controls and constraints can be implicit in the access control table information, such as the need for a person to be in two particular NTKC's before having access to a resource, or conversely having access only if the person is in one, but not both, of the NTKC's. Access groups which contain implicit information of this form must be carefully considered in the design of the routines which are to be utilized for updating the access control tables (to be considered in section 3.2.4). The security classification levels of Top Secret, Secret, and Confidential can also be easily handled by implicit considerations in creating the table entries. If an object requires a TS clearance level for access, only those persons can have this object included in their access lists. This implicit approach has one major side-effect, namely that subjects other than persons (e.g.,

terminals) have unnecessarily complex access table entries. Instead of explicitly stating that a terminal is capable of TS-level operation, or TS-level for NTKC's i and j, the table must list all such objects as being accessible by the terminal. This complication seems unnecessary and deserves additional effort to resolve.

The security clearance levels form a hierarchy in that Top Secret implies Secret which implies Confidential.* This same hierarchical approach has been applied to capabilities by Garwick (GAR-73), but might also apply to objects or subjects if hierarchical relational structures exist of the general form A implies B implies C, etc. Such structures might very well apply to a set of files or members of a large project team. (The reader should recall the objection to hierarchical authorization structures by Wulf, et al (WUL-73) as discussed in Section 2.2.5).

The detailed design of the program for the Security Controller must carefully consider these aspects of data storage compaction and the corresponding retrieval and update processing requirements. The three-dimensional nature of the information can readily "explode" into a very large array of data at some future date when a large number of users are to have controlled access to individual objects (HOST's, files, etc.) with any of several capabilities. Even if the initial SC usage is limited to a subset of these axes, the design should be based on a structure which will support growth without requiring a subsequent major redesign or restructuring of the data.

*This hierarchical relationship has not universally been accepted as being factorable across NTKC's; e.g., a person may have Top Secret access to NTKC #i, but be limited to Secret for NTKC #j. This may complicate the generation of the access table entries, but does not affect its usage.

A flexible organization which meets all of the known requirements for the SC's access control function has been defined by Kaufman (KAU-74), and is summarized below. This approach has particular merit in that it can efficiently handle both the simple initial usage and more complex future usage in an open-ended manner.

The basic profile of a requestor (subject) would be stored in a block of information which would also include access authorization in terms of (object, capability) pairs. Considerable factoring, and hence compaction, is possible as indicated in Figure 3-2. Item "a" in the figure shows an entry which is a direct (object-capability) pair, which might simply represent access to a particular HOST machine. In contrast, the second entry in the access authorization table is a pointer to "b" which is an arbitrary capability list for an object which, in this example, is owned by this subject. The third entry is also a pointer, but in this case to an access group. Another subject is also shown to be a member of this group. The fourth item is another direct representation, and the fifth is a pointer to "d"; an access group in which a further level of factoring has been applied, i.e., with several objects sharing the same capability list.

The example situation depicted in the figure is intended to illustrate the open-endedness of the approach, rather than suggesting that compaction by means of the various factoring schemes is necessary. We shall discuss the aspects in which factoring both helps and complicates matters in subsequent sections of this report.

A rather subtle issue that must be considered in the design of the SC is whether the access table should be structured by subject or by object, i.e., the factoring could be either:

<subject> (set of <object>, <capability> pairs)

or

<object> (set of <subject>, <capability> pairs).

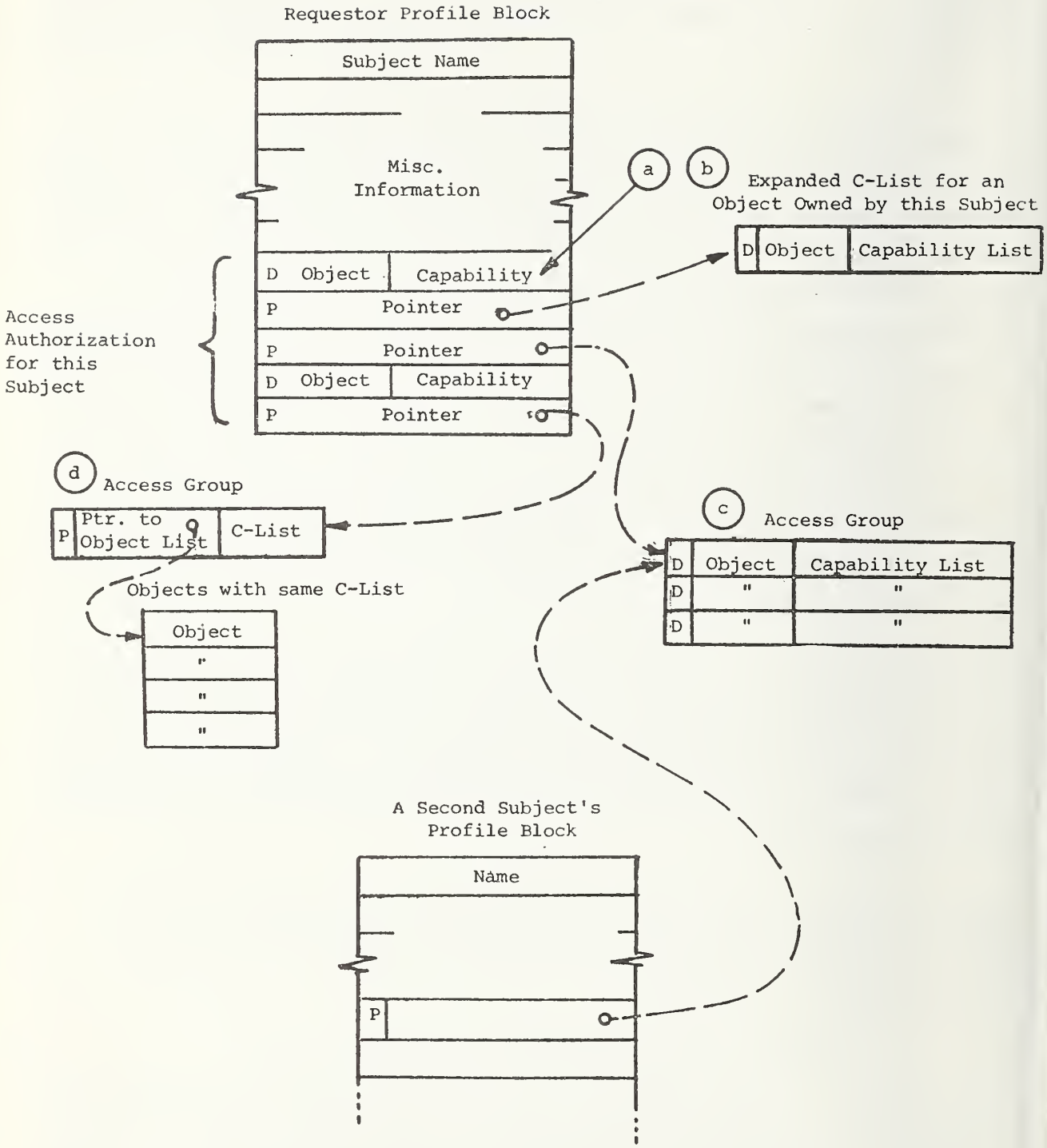


Figure 3-2. An Example Requestor Profile

Any access request will include both the requestor (subject) and the resource (object), so both approaches are feasible. Structuring the data by subject has benefits in that;

- The SC must look up some subject-oriented information anyway (such as identification/authentication data).

- The subject may simultaneously request access to two or more objects, in which case factoring by subject would be more efficient.

- Adding or deleting subjects is easy.

- A subject's request of "what access privileges do I have" is simple to respond to.

Conversely, an object-oriented structure has similar benefits when new objects are added or deleted, or when one desires to find out what subjects have access to a given object. These tradeoffs can only be made if one knows the relative frequency of the subject- versus object-oriented actions. However, the first two subject-oriented benefits do not have counterparts in the object-oriented approach, and hence lead us to recommending a structure that is factored by subjects.

The organization of the access table should consider both the usage aspects and the need for updating the entries. These are typically conflicting requirements, which must be resolved based on operational usage. Our analysis has emphasized the need for efficient usage since updating is much less frequent, and if necessary can be performed in an off-line mode of operation.

3.2.3 Usage of the Access Control Table

When an access request is made, the SC must retrieve the information block containing the requestor's access capabilities (we will assume the form of Figure 3-2), and then must search this linked list to determine if the access is authorized. The particular method of search is left as a design issue, which should be based on the level to which access controls are to be applied by the SC, and the tradeoffs between search and update considerations.

Processing is also necessary whenever the composite subject (e.g., person, terminal, and process) is not in the table as a single entity, but rather exists as separate subjects (as discussed in Section 3.2.1). The stored information would be retrieved for each entity of the composite requestor.

If the request is for a given person to access a particular object, the desired information is the capability vector associated with that (subject-object) pair. This requires a search of the subject's access list; i.e., the computer representation of the (object-capability) plane of the access matrix that corresponds to that particular subject. The search must find if the object of interest is present, and if so the associated capability vector must be extracted. The capabilities of the terminal and any other associated entiti must then be determined and processed with those of the user to determine the composite capability vector. If the only level of access control is a GO/NOGO decision as to whether a connection should be established, this vector need only be a single bit in length. However, the structure should be kept more general for future growth potential, e.g., such that capability information can be sent to the serving HOST.

The information that would be sent would be that portion of the requestor's profile applicable to that resource, e.g., the capabilities for that (subject-object) pair. A mapping of this capability vector would then have to be made at the serving HOST to convert from the standard form of the SC to that of the particular HOST.

When the requestor (subject) and resource (object) are in separate SC domains, several new options must be considered for the access control matrix. One approach is to include all relevant subjects in the matrix that controls a particular object, which provides the object-oriented control that is felt to be needed. In effect, the object information is centralized at the SC responsible for that object, while the subject information is distributed; i.e., the access capabilities of a given user may be scattered across two or more SC's, such that each SC handles its own local users.

The alternative is to centralize the subject information, e.g., at the requestor's SC, and to distribute the access information for a given object. This latter approach has two major disadvantages: (1) the access checking is performed at the requestor-SC instead of the resource-SC so that the desired object-oriented protection is not provided, and, (2) any necessary global searches related to a given object are very time-consuming since the storage is subject-oriented at each SC. Therefore, the recommended approach is to include local and remote subjects in the access control matrix of the SC that protects a given object. The subject-profile that would be sent to the object-SC would include identification, clearance, and NTKC information, but would not include any new entries to the access control matrix.

If a given person is to be deleted from the list of subjects, that person would initially be deleted from his local SC, which would "break" his ability to access any of the network objects to which he previously had authorization. These remote SC's would be updated by an SC-to-SC message to purge the particular user from the access control matrix; a simple task since each SC has this information organized by subject. Either all possible SC's could be notified in this manner, or a table of relevant SC's could be kept as part of the user profile.

3.2.4 Updating the Access Control Table

There are two major issues associated with how the access control tables should be updated: (1) whether it can be performed efficiently in an on-line mode, and (2) how to control the process. Each of these areas will be considered in the following paragraphs.

3.2.4.1 Feasibility of On-Line Update. Assuming that the access information is handled as a set of linked lists, the update process involves searches, deletions, additions, and changes of blocks of information and pointers between these blocks. On-line update of this information appears feasible if a sufficiently dynamic storage management scheme is utilized, but the impact of such a scheme must be carefully considered in terms of its search speed and any complexity that it adds to the Security Controller (e.g., for proof of correctness considerations).

Updating becomes a matter of modifying the linked list representations of the (object, capability) pairs, which could be performed either on or off-line. By off-line updating, we mean a batch-processing method of regenerating the tables on a periodic basis, utilizing either the SC itself or some other machine for such processing. Conceivably, this processing could be running as a background task on the SC during its normal operation, but this would add complications well beyond the benefits. Usage of a redundant (stand-by) SC for such updates would be more feasible, but is not necessarily consistent with the optimal usage of such redundant equipment (see Section 3.7.2). Therefore, if off-line updating were chosen as the desired method, such updates should be performed on an auxiliary machine.

On-line updating would require additional SC programs and would require that storage be allocated dynamically or that sufficient reserve space be provided in advance for the updates. The linked list approach of Figure 3-2 might utilize a combination of these two schemes, with a pre-allocated profile block which could then be updated to include either direct authorizations, or pointers to (object-capability) lists such as access groups. In the initial mechanization, only the pre-allocated portion might be required, thereby

deferring the need to actually implement the dynamic portion. However, the basic design would include the ability to expand via the dynamic allocation scheme.

3.2.4.2 Controlling the On-Line Update Process. The minimal requirement for on-line update would be for a simple debug-like facility to allow changes to specified memory locations. However, changes to information within the SC must be made under very tightly controlled circumstances due to the potential consequences of erroneous profiles, etc.

One approach to providing this control is to only allow changes to be made from a single terminal (and perhaps by a single person) which can provide an arbitrarily high degree of physical and procedural protection. This centralized approach has a certain intuitive appeal, but is vulnerable to errors in the administrative pipeline that would feed change requests from the user community to the person responsible for making these changes, and also looses the "reasonableness checks" inherent in a distributed scheme in which updates would be made by persons involved in the activities.

The basic problem with distributed updates is the need for selective update controls. The selectivity aspects include the need to control that:

- only the owner of an object can grant, modify, or remove capabilities to that object.
- these capabilities can only be granted to subjects that have appropriate clearance and compartment requirements.
- the only capabilities that can be changed are those related to the owned object.

The needs can be met by the usage of a "trusted" routine that performs the updates on one's behalf after having made the necessary checks. A security officer would serve this role in the centralized approach, while a special SC update routine would be required in the distributed scheme. This update routine would itself be a subject, and in fact would be the owner of the access control matrix. As such, it could "bootstrap" the creation of the matrix from the initial triple, (update routine, access matrix, owner), to whatever current state the matrix may obtain. All requests for change would be via "messages" sent to the update routine, which would then check the authorization before actually making the changes.

3.3 THE SECURITY CONTROLLER MECHANISMS FOR ESTABLISHING A CONNECTION

If the requested access of a subject to an object is authorized, the SC must then create a working connection between these two parties. The notion of "creating a connection" involves several levels of protocol, and therefore is dependent upon the physical and logical organization of the network. However, we will assume that the following functions must be performed in any network, and will address the issues related to these aspects.

- Control over the initial requestor-to-SC connection.
- Determining the path for the control messages that are used to create a working connection between the requestor and the resource.
- Handling exceptional conditions on a set-up attempt.
- Crossing inter-network boundaries (gateways).
- Initial control message contents, e.g., should requestor profile information be sent?

3.3.1 Control Over the Initial Requestor-to-SC Connection

One of the requirements established in Section 2.2.1, was that the SC perform a Reference Monitor-like function, including the concern that it be always invoked. To ensure that this happens for each initial access request, one must force all such requests to be made via the SC, and can enforce this policy by taking advantage of the restrictive aspects of the cryptographic equipments, i.e., they will only pass meaningful information if the two ends have matching keys. Therefore, we can ensure that any requestor must initially contact the SC by setting the SC's initial crypto settings to some known condition (e.g., null), while the initial crypto settings of each resource would be known only by the SC.

The recommended requestor-to-SC "handshake" for their initial connection would be as follows. First, the requestor would "activate" a physical connection between itself and the SC, and would then send an identifying message of the form, "Hello, I'm device # xxx," to which the SC would respond by setting up a temporary set of working keys to protect any subsequent requestor-to-SC dialog. (If a quick test of device authenticity is desired, an echoed message can be utilized to ensure that both ends have matching cryptographic keys. This and other aspects will be considered in Section 4 which will discuss the cryptographic devices.)

3.3.2 Selection of the Path for Set-Up Control Messages

When the SC has determined that a requestor is valid and that the particular request is authorized, it must then set up a protected working connection between the requestor and the resource. The principal aspect of this is the insertion of the working keys into the cryptographic devices at the two ends, which would be performed by sending a special control message to each device.

There are several possible paths for the distribution of these set-up messages One approach takes advantage of the fact that the SC is already in contact with the requestor, and therefore relays the setup message to the resource via the requestor. This minimizes the number of connections which must be established, but requires that the SC-to-resource message be protected from the requestor during the relay process.

A second approach is to relay the control messages in the opposite direction, i.e., from the SC to the resource to the requestor. This requires the establishment of an additional connection, and merely reverses the roles of the requestor and resource in terms of protecting the set-up information to be sent to one via the other. However, it does provide a path for SC-to-resource information that does not involve the requestor (e.g., for profile information). A third alternative is for the SC to send the control messages directly to both the requestor and to the resource, and then rely on the two entities to establish the desired connection, since their cryptographic devices have been "primed" with identical keys.

The tradeoffs related to these three alternatives are a function of the cryptographic and communications network levels and will be deferred to those sections of the report. For our purposes here, we will assume that these control messages are distributed to the cryptographic devices by a means appropriate to a given network.

One other aspect of the set-up control messages is whether the SC should receive any feedback regarding whether the connection was actually established. This is probably necessary for two reasons: (1) to ensure that audit records regarding the connection creation are correct, and (2) to help in resolving exceptional conditions which may be occurring during the attempted connection creation. These latter conditions will also be the subject of the next section.

3.3.3 Handling Exceptional Conditions on Connection Creation

Any of several things may go wrong during an attempted connection creation, and these anomalies must be dealt with by well-defined procedures. The principal concerns are to categorize these conditions into generic problem areas to ensure that all such problems have been considered in the recovery procedures. The generic problem areas include the following:

- All or part of the set up messages get lost or otherwise become undeliverable, leaving the connection in an unknown, partially made state.

- The content of the delivered message is in error leaving the connection in an "unbalanced" state.

- The resource is unavailable; e.g., either dead, running at a different security level, or otherwise not accepting requests. Note that if a HOST changes its security level it could either notify the SC immediately or could wait until a connection attempt was made for such notification.

- If connection sequence numbers are utilized as protection against the "record and play-back" of connection establishing messages (analogous to Baran's Pre-Filtering Key (BAR-64)), then an error can occur if these sequence numbers get out of step.

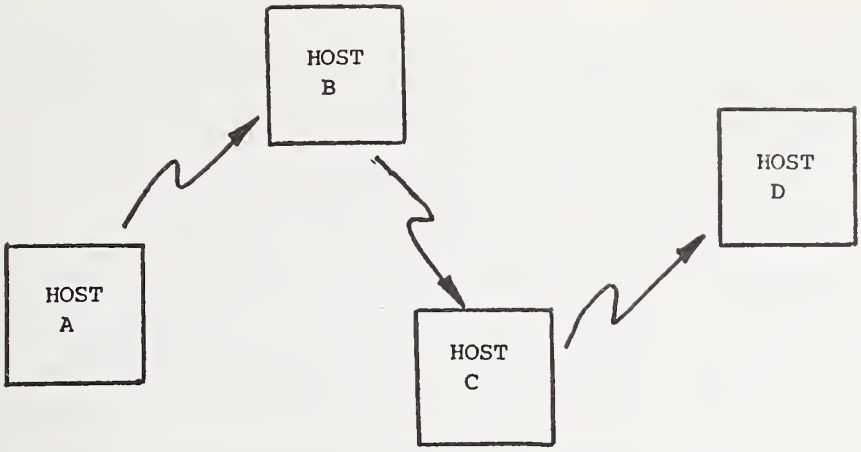
The above list is not necessarily complete, and one should expect that the recovery procedures will have to be extended from time to time as new error categories emerge.

3.3.4 Crossing Inter-Network Boundaries (Gateways)

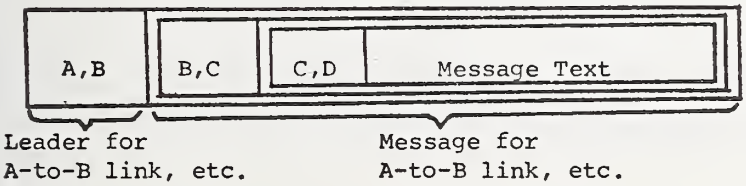
Addressing messages across network boundaries introduces several new problems including the requirement that a more global naming convention be established. The basic requirements of such a convention are that the names must be unique and that there must be a way of mapping the names to particular entities. The most common approach would be to utilize a hierarchical naming convention of the form <network>·<subnet>·<HOST>·<process>, in which higher order qualifiers may be implied (e.g., in the manner that an area code is only dialed when needed). The choice of utilizing the implied information results in variable length names which must be "parsed" to determine their meaning, but does reduce the size of the names that must be transmitted (on the average). This variable format also yields an open-ended addressing scheme which provides considerable growth potential, but does introduce additional complexity. A second aspect of this complexity arises when one considers the need to include both the source and destination addresses in the header portion of a message. However, the same parsing scheme could be applied to each address so the problem is merely an extension of that of variable length addressing.

One of the primary counter-arguments to keeping the addressing method simple is the need (or potential need) for multiplexed cryptographic devices, which would select the proper key based on the addressing information. These devices must be kept as simple as possible to ensure their accreditation as being secure. We will leave this area of fixed versus variable format addressing as an open issue to be resolved within the context of a particular design.

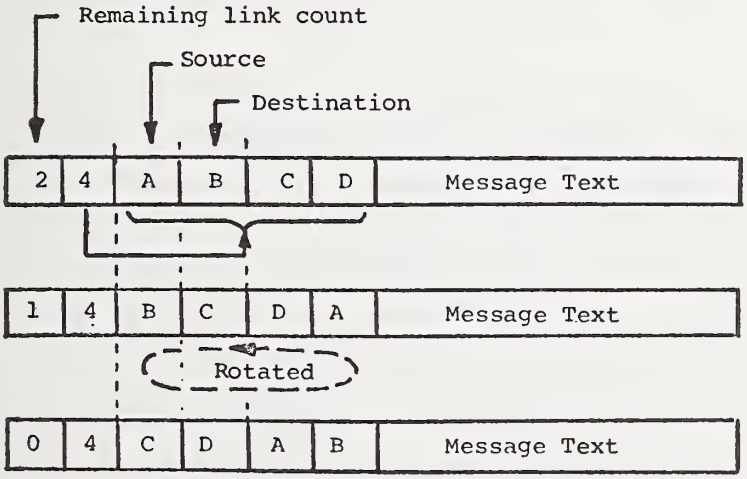
Knowing only the source and the destination addresses implies that some form of routing information is implicit in the network. This is typically the case within the communications net, e.g., store-and-forward switches with routing tables, but is not necessarily so for higher-level entities as shown in Figure 3-3(a), in which information may be carried along as part of an extended message leader. One approach is to embed (or nest) leaders as shown in Figure 3-3(b). Each outer leader would be discarded after use.



(a) HOST-Level Routing of Messages (e.g. for N-th Party or Gateway Functions)



(b) Nested or Embedded Leaders



(c) Circular Leader

Figure 3-3. A Comparison of Linkage Addressing Schemes

A second approach which has been adapted from Farber (FAR-74) is shown in Figure 3-3(c). In this scheme, all of the intermediate nodes are carried in the header, and are circularly shifted at each node which having been the destination, now becomes the source (except when the link count goes to zero, at which time it is known to be the ultimate destination). The method avoids the need to repeat information as was done in the nested leader approach and also retains a "trail" which can be utilized for return messages. In each approach, the message content may be a combination of information to be acted upon at the various intermediate nodes as well as at the ultimate destination, but our only concern here has been with the addressing. Other considerations must include such things as how control information is passed, which may not map readily. End-to-end protection of communication paths which involve a gateway between two nets may also be a problem, depending on several factors including:

- Whether the nets utilize identical Security Controller and Intelligent Cryptographic Devices.

- Whether control information (other than leaders) is passed in the clear or is enciphered.

- Any other transformations which involve the text of the message.

One straightforward solution to the problems of gateways is to consider each gateway as an intermediate HOST which happens to reside in two networks. In the general case, one of these nets is a local net while the other is the global network of gateways that interconnect local nets. In special cases, the gateway might directly connect two or more networks without usage of a global net.

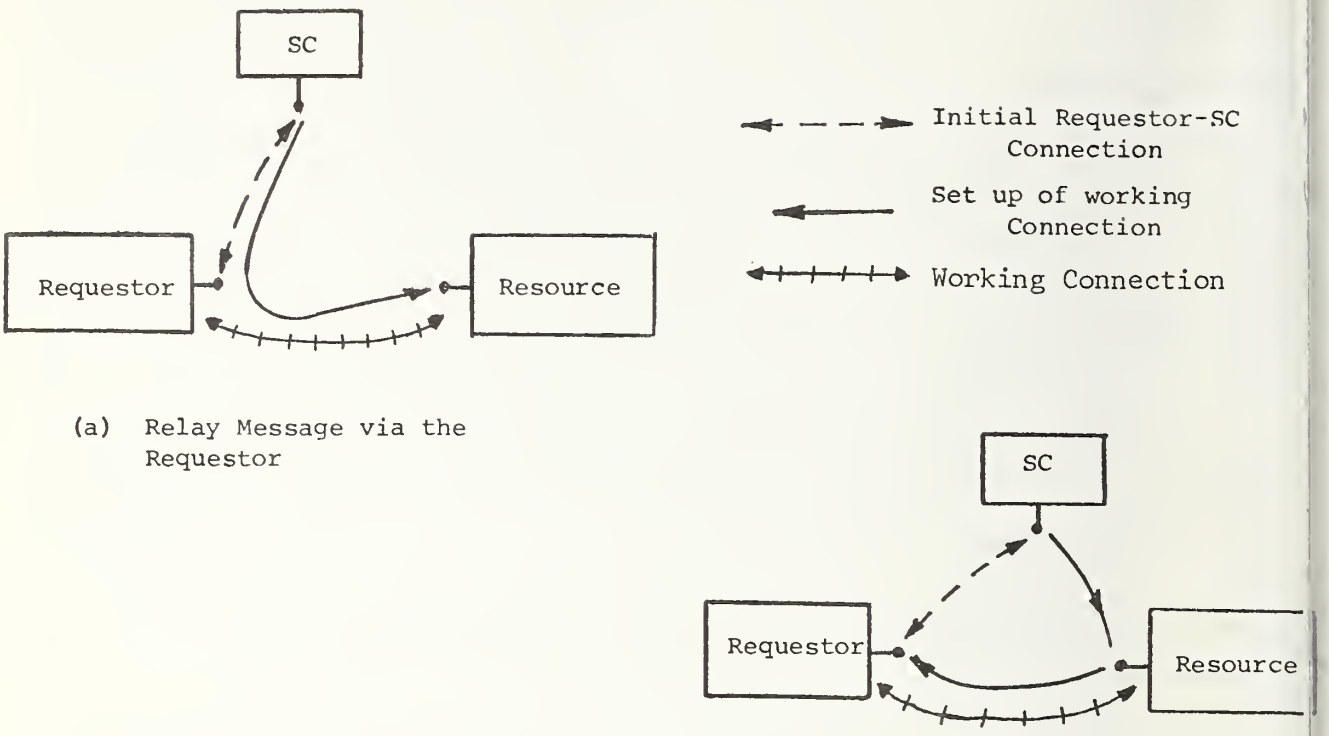
The detailed aspects of gateways and networks of networks are beyond the scope of this current effort, but must be considered in the design of the SC to the extent that open-ended addressing, protocol mappings, etc. influence its evolutionary development.

3.3.5 The Contents of the Initial Control Messages

When the SC establishes a working connection, it must transmit certain control information including addressing and control information for the communications network connection, the working keys for the cryptographic devices, and any necessary HOST-level data. For the case of relay messages, the addressing information can be handled by either of the two methods that were considered under gateways, namely nested leaders and cyclically permuted leader fields. For direct SC-to-requestor and SC-to-resource messages, both addressing methods degenerate to a simple source-destination scheme, but for the creation of a connection, information must be sent to two or more entities, and the selected linkage addressing scheme would come into usage to set up the proper initialization of the connection. An alternative approach would be to "prime" the two ends by means of independent SC-initiated messages as shown in part (c) of Figure 3-4. Parts (a) and (b) of that figure show the two "relay message" approaches for comparison. The tradeoffs related to these three schemes involve several factors including:

- The communications net. What overhead is required to initiate a connection, and how important is it to minimize the number of such connections. (Method (a) requires that two connections be established, while (b) and (c) each require three.) These factors will be further explored in Section 5.

- The relay mechanism. For economic reasons, the relay mechanism may only be associated with certain multiplexed devices (e.g., HOST computers). In that case, the relay operation would be performed via the resource as shown in (b), rather than requiring each requesting terminal to have a relay capability. These issues will be further explored at the cryptographic device level in Section 4.



(a) Relay Message via the Requestor

(b) Relay Message via the Resource

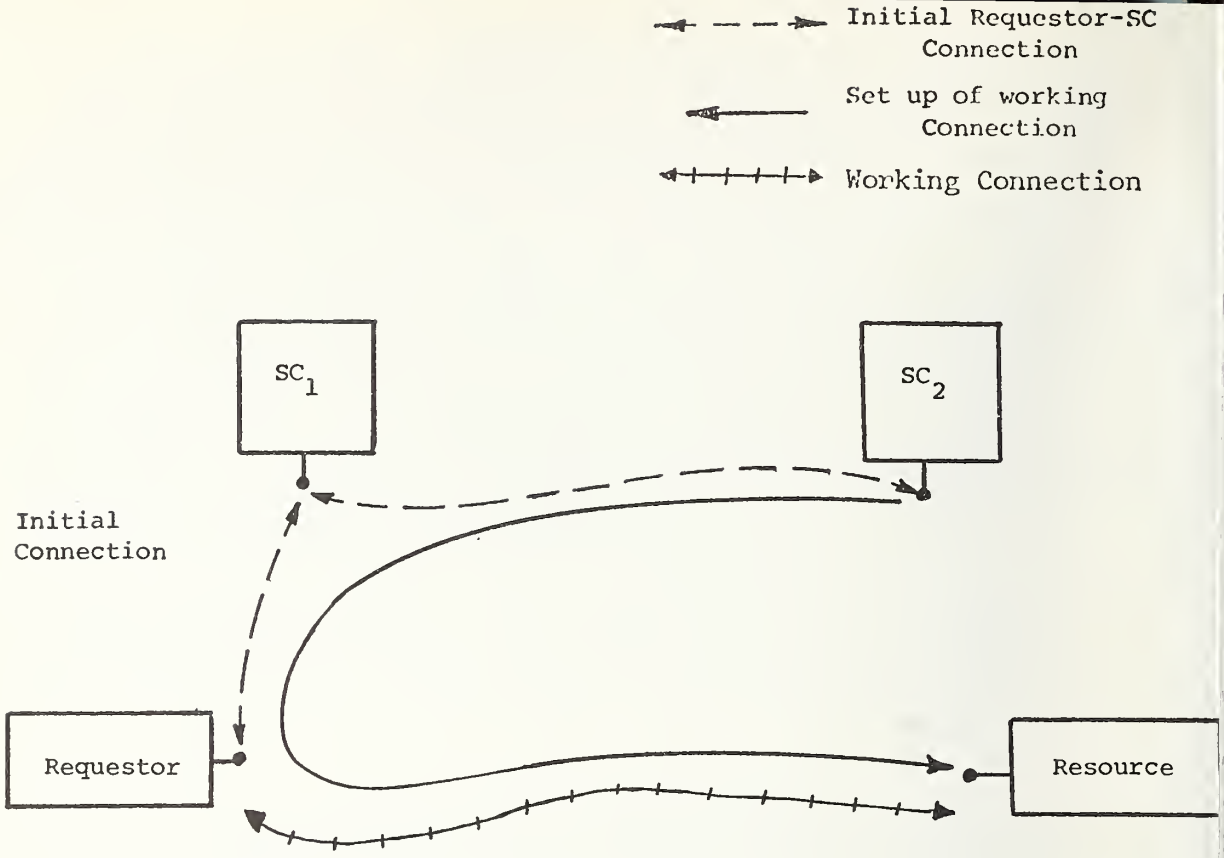
(c) Separate Priming of Each End

Figure 3-4. Alternative Methods of Establishing the Working Connection

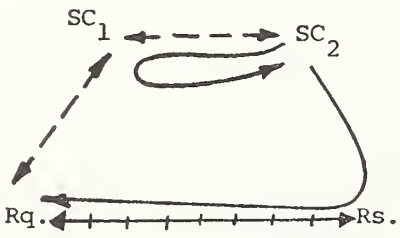
- Handling of Inter-Domain Connections. When two SC's are involved in the creation of a connection, the approaches of Figure 3-4 expand in complexity and in the number of possible combinations, since either SC_1 or SC_2 can perform the distribution or some combination of the two can be involved. Following the rule of Section 3.2, the SC "guarding" the resource will be responsible for establishing the connection which reduces the viable combinations to those of Figure 3-5, in which the three variations correspond to those of Figure 3-4. In part (a) of Figure 3.5, the relay message is sent via SC_1 which modifies the remote keying information since SC_2 does not know the private keying variable associated with the requestor. Similarly, this function is performed in (b) prior to sending the connection creation message via the resource, and in (c) by relaying through SC_1.

- The need to notify the SC if the connection attempt was successful. For audit reasons, the SC should know if an attempted connection creation was successful or not. One approach to achieving this is to relay the connection request one additional link; namely to echo it back to the SC with connection status information indicated in the message field.

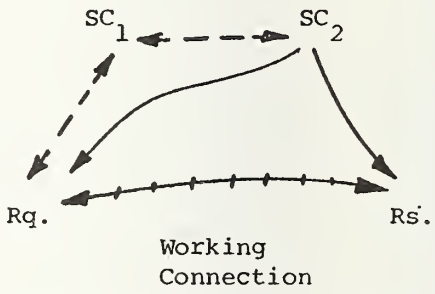
The information in the relay mode of operation must include that of several protocol levels as described above and as shown in Figure 3.6. One of the concerns for the HOST-level messages is how profile information should be sent from the SC to a resource, since it represents a potentially awkward mixing of the HOST-level protocols. That is, if the profile data were to be sent as part of the initial connection message, it would be mixed in with control messages, and thereby would require a "sorting out" of the message contents



(a) Relay via SC_1 and the Requestor



(b) Relay via the Resource



(c) Separate Priming of Each End

Figure 3-5. Connection Creation When Two SC's Involved

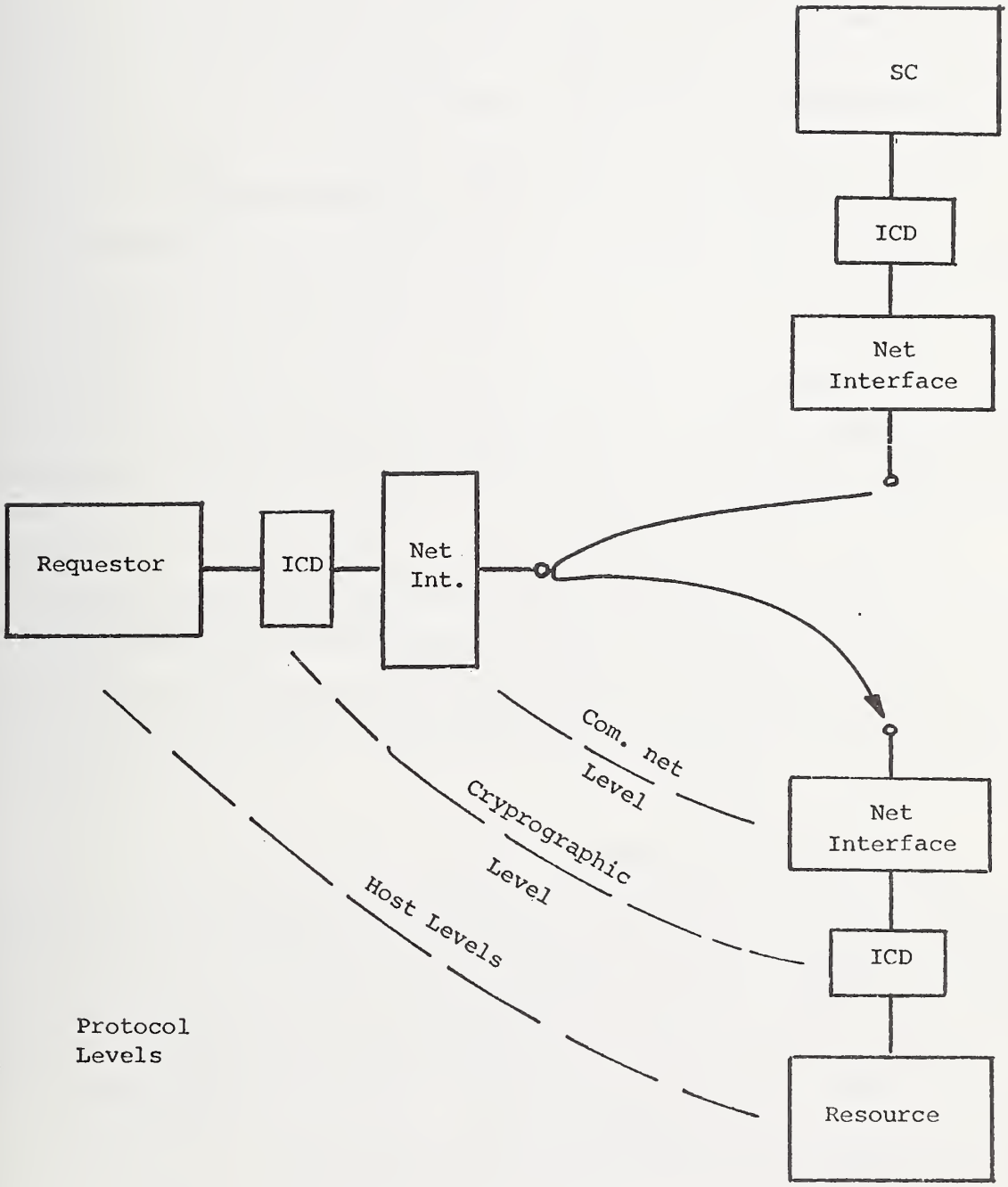


Figure 3-6. Levels Involved in the Connection Creation Process

for different semantic and syntactic interpretations. A better approach from the point of view of keeping the design clean and straightforward is to break these two needs into separate messages, i.e., having the resource contact the SC to obtain the profile information when it learns of the need for this data. The disadvantage of this scheme is the need for an additional connection to the SC (i.e., by the resource), but this should result in minimal delay since all operations would be computer-generated.

3.3.6 Control Over Play-Back of Connection Creation Messages

Protection must be added to preclude the unauthorized establishment of a connection via play-back of a recorded (legitimate) connection creation message. This requires that some aspect of the connection creation sequence be different for each connection, such that a previous message would not be valid for subsequent usage. Either the initialization or the message content could be modified, e.g., using a different initial encipherment or a connection sequence number respectively.

A variation of this same threat would be to attempt to deny service on a legitimate connection by playing back a connection creation message to one end of an existing connection. This threat should be precluded in the design by only accepting such connection creation messages when the mechanism is in an initiate mode of operation.

3.3.7 Implicit Connection Creation

There may be circumstances in which certain terminals require that predefined working connections be created implicitly by an "off-hook" operation at the terminal. This operation would give the effect of a dedicated line between the two entities, and as such would not be subject to SC approval or disapproval. However, the crypto keying might be by means of the normal SC-initiated key distribution (with an implicit authorization), and the SC could thereby maintain audit information on the usage of the connection.

3.3.8 Connections for Broadcast Messages

If two or more recipients are to receive a given message, either a single broadcast message can be sent to be received by all such sites, or the message can be repeated for each recipient. This choice must depend on the message protocols and communications media involved. The addition of network security mechanisms such as the SC and ICD's introduces a new constraint on such messages, namely all recipients must be approved by the SC and all of the appropriate ICD's must be keyed for the receipt of such messages. If broadcast messages are feasible and desired in a given net, the SC must be extended to handle the functions of multiple destination authorization and keying.

3.3.9 Connections for Unclassified Work

Network access for usage at an unclassified level could be possible if a given HOST machine allowed mixed operations of classified and unclassified work, or if the network includes both types of HOST systems. The SC could be utilized to establish an unclassified connection by creating a null key for the dialog, or if such usage is particularly common, such keying might be allowed by a manual operation at the terminal/ICD. The set of classified HOST's and the set of unclassified HOST's could share a given communications net, but remain logically (and securely) separate by means of the crypto keying. Only if one or more HOST's appeared in both nets simultaneously would the two nets have any overlap of concern.

3.4 THE SC/HOST-LEVEL MECHANISMS FOR CONTROLLING CONNECTION USAGE

Once a working connection has been established by the SC, its usage is primarily under the control of the requestor/resource ends and their respective cryptographic devices. Protection is needed against several forms of misuse which will be discussed in the following paragraphs.

The HOST's can misuse a connection by multiplexing multi-user traffic over a connection established for a single user, and can also mis-label the sensitivity of information sent over the connection. These problems must remain as sole responsibilities of the HOST computers. However, the network can provide some added controls in the area of accidental misusage, since checks can be made on the validity of addressing and classification fields relative to the values initially set up.

Other forms of protection can also apply against the play-back of recorded cipher-text messages. If such messages are cryptographically self-synchronizing they can appear as valid messages, but protection can be added by a combination of HOST and cryptographic level checks such as connection or message sequence numbers, time-stamps and check sums (to detect maliciously inserted bit changes). The use of sequence numbers introduces a new level of synchronization which must be considered. Several forms of usage control are required for recovery from errors such as the loss of a message, a message arriving when not expected, and crash-generated problems in an operating system. These problems have some security side-effects (e.g., possible loss of cryptographic or message sequence synchronization), and thereby can have operational impacts.

When the usage of a connection is completed, the SC should be notified so it can "close-out" its audit record for that request. Questions arise as to which end of the connection should notify the SC, or should both, particularly if they reside in different SC domains. Other questions relate to whether the notification of completion requires the full identification/authentication prelude, and whether some special SC action is necessary to ensure that the cryptographic keying for the connection is reinitialized.

3.5 SC/HOST-LEVEL MECHANISMS FOR MONITORING

The SC will need to maintain certain status information for each requestor that is in some state of requestor-to-SC dialog, or that has an outstanding connection which the SC has created. This information is essentially all the SC knows about a particular request, and can become the basis for the SC's

audit record. Other information could also be included such as whether any erroneous password attempts were made (prior to stating it correctly), and whether any unauthorized requests were made.

Certain threshold conditions should cause an immediate abort and/or alarm for an abnormal requestor-to-SC dialog such as mis-stating a password three times in a row. Other monitoring tests require more of a historical perspective to determine if a penetration attempt has been made, and could best be performed in combination with other information that might be maintained in a Network Security Center which would be the focal point for the analysis of SC and HOST-gathered audit information. More detailed investigation is required to determine the viability of such a center (or set of centers), since there are open questions related to:

- How to combine and correlate the audit information from the various sources.

- What network performance degradation occurs due to transmitting the audit information.

- What procedures should be followed to ensure the validity of the audit data, e.g., whether HOST-to-Net Security Center connections should be set up via the SC.

- Where audit data interpretation should be performed, e.g. central or distributed checking.

- What audit interpretation tools are required, and in particular what new tools are needed due to the distributed nature of the source of audit information.

The need for some integrated audit interpretation seems necessary due to the difficulties in interpretation of the otherwise fragmented audit trail which may involve two or more SC's, multiple HOST's, etc. However, we will leave this as an open issue that requires additional attention in the detailed design phase of the network development.

The Network Security Center would also be a logical candidate to handle the coordination functions related to updating the SC mechanism (e.g., code changes), for recertification as required, for updating cryptographic device "private keys," etc.

3.6 SECURITY ASSURANCE ASPECTS

The network security assurance functions fall into four major categories: (1) certification of the SC mechanisms, (2) handling of SC data such as profiles, (3) self-checking, and (4) the necessary physical and procedural controls. These areas include both initialization and on-going aspects, which must be considered in the design and operational usage of the mechanisms.

3.6.1 Certification Issues

The principal impact of the certification requirement is on the system design, since any after-the-fact attempts to demonstrate system integrity would tend to be fruitless. The entire design process should be oriented towards the use of the best available methods of program development (e.g., structured programming and chief programmer team approaches) as well as ensuring that additional security needs are also met. For example, in most programming environments, a program is considered to be correct if it does what it was supposed to do. Due to the security implications of the SC, its programs must do that, and only that, as well as having well defined failure modes (e.g., fail-secure operation). Therefore, the need for certification causes the following design constraints. First, the selection of an implementation language must be based on the abilities of the language to support structured programming and proof of correctness techniques. This language should also serve as a convenient and expressive tool for design and documentation, since one aspect of accreditation

is that one or more persons be able to thoroughly understand the entire program. Other aspects of the language selection will be considered in Section 3.7.

A second design constraint imposed by the certification requirement is that the system remain secure even under most failure conditions. This requirement involves substantial checking capabilities which are not available in current computers of the size envisioned for the SC (e.g., a minicomputer) and therefore, implies that redundancy and checking must be applied externally (e.g., by duplication of equipments and cross-checking). The existence of such redundancy leads one to consider a degraded mode of operation in which only one set of equipment is used, i.e., without checking. If several other levels of checking and controls have been applied, such operation may be feasible, but must be considered in detail within the particular context of the application and equipment being utilized. A more detailed description of the tradeoffs involved in such usage will be discussed in Section 3.7.

3.6.2 Handling of SC Data

The issues of how access control information should be entered into and retrieved from the SC were addressed in Section 3.2, and will not be repeated here. One also needs to consider the handling of other security data such as the creation of new subject profile blocks, i.e., the basic kernel of information about a given person (or other subject). However, these needs are basically the same as those for access control, and involve the issues of centralized versus distributed updating, and what physical and procedural controls are required for these processes.

3.6.3 Self-Checking

The integrity of the SC mechanisms can be ensured, to a given level of confidence, by the usage of built-in checks. These checks can be applied in several forms, differing in the manner in which they attempt to detect anomolies and whether they are executed concurrently with the regular processing or are interspersed at given intervals. Redundancy checks can

be in either category depending on whether the equipment is replicated, or if it is utilized repetitively (e.g., make each computation twice to check for transient errors), but in each case, the attempt is to detect errors by checking the results of the process of interest. In contrast, diagnostic checks test the basic equipment to ensure that it is functioning (again to some level of confidence). These checks do not necessarily relate to any particular process usage of the equipment, but additional tests could be added to simulate a given process (e.g., with predefined data and results).

A third category of self-checking is that of pseudo-penetration tests in which a set of "canned" attempts would be made to break the SC's security. These tests could be made via processes running inside the SC (e.g., in a background mode of operation) or could be simulated externally (e.g., by an attached "box" which would play the role of the penetrator). The latter can be considered to be self-checking if the "box" is a part of the SC, but the concept would still apply even if it is a remote device or one that is periodically (or at random intervals) connected to the SC.

3.6.4 Physical and Procedural Controls

Network security requires that substantial physical and procedural security controls be applied to the entities (devices and programs) involved in the net. These controls must cover all such entities and must apply at all time epochs including their initial development, certification when installed, and re-certification after any modification. The need for controls even during the development of the entities should be stressed since it is easy to overlook the need to certify production tools as well (e.g., the compiler used to generate executable object code from a certified source program).

Particular concerns of this investigation are the needs related to the SC, and therefore we will focus on those needs instead of considering the well established HOST-level security controls. The SC can be expected to run in a relatively unattended manner. It may need occasional replacement of magnetic tapes (for audit data collection) although this need might disappear

if the SC were to send audit information directly to a Network Security Center. Similarly, the Security Officer might be at the SC, or at the remote Network Security Center, but in either case could monitor the general behavior of the SC to detect a possible system failure, etc. Recovery from such failures via remote operations are technically feasible, but may have significant security considerations that must be carefully resolved in the detailed design. It may also be desirable to occasionally reload the SC program to "break up" any compromise situation that may have occurred; by either accidental or malicious means. This would present another concern as to whether such operations could be performed securely via remote methods.

3.7 OTHER DESIGN ASPECTS

Two forms of control programs must be considered in the design of a secure net along the lines that have been developed here. First, is the need for some form of network control program (a set of processes that will control the flow of information across a connection between two network entities) and secondly, there must be a security control program which establishes these connections when authorized. The two are interrelated in the sense that the HOST's must be able to communicate with the Security Controller, as well as with each other. However, we will discuss the two aspects separately. We will then define the error control aspects related to the control program of the SC, and finally, describe the hardware requirements that are necessary to support the SC functions. The extent to which the security mechanisms impact performance will also be considered.

3.7.1 Network Control Programs

When two or more computers are to be interconnected into a network, they must be provided with a mechanism by which they can communicate control and data information. This mechanism can range in complexity and capability from a simple "terminal look-alike" scheme, to a full scale operating system-to-operating system interface as utilized in the ARPA network. With the terminal look-alike scheme, there is a minimal impact on the HOST operating systems since each thinks that it is talking to a set of terminals, but the

operations which can be performed are limited by the command and data constraints of the terminal-oriented transactions. At the other extreme, there is a very significant impact on the HOST operating system, but with a high degree of generality in the resulting information flow. Intermediate solutions typically involve taking a "peripheral look-alike" approach, which provides a middle-ground in terms of operating system impact and generality of usage.

Like so much computer terminology, the notion of a Network Control Program (NCP) has taken on a variety of meanings, ranging from a particular level of HOST-to-HOST protocol which was its original meaning, to the entirety of the network software, which is its more popular, current usage. Since the term has become broadly, and rather vaguely defined, we will not attempt to re-define it here, but will use it as a generic term for the HOST-related network software. The discussion will generally be limited to ARPA-like protocols (CRO-71), but will also consider suggested variations such as Walden's message-switching protocol (WAL-72).

Many different approaches were taken in implementing NCP's for the ARPA net, with significant differences arising in the NCP-to-operating system relationship (MET-72). These ranged from distributing it across the OS, to having the NCP as an integral part of the Operating System (OS), to having it run as partition under the OS. However, in the latter case the NCP was still executed in supervisor mode, and was interfaced to pre-screen supervisor calls and I/O interrupts to determine if they were network-related.

The insertion of Security Controller and Intelligent Cryptographic Devices into a network (as shown earlier in Figure 3-1) can have either a small or an extensive impact on the NCP's of the HOST machines, depending on the extent to which these changes are kept transparent to the original operation. For example, if the HOST-to-ICD interface is designed to "simulate" the original HOST-to-Network interface, and if the added requirement of contacting the SC for connection authorization is implemented in this interface, the NCP changes will be quite minimal, although the operation might tend to be

rather inefficient and awkward. A better long-term solution would be to accept the rather significant NCP changes required to handle the SC-related dialog and to move the message handling functions (e.g., retransmission and flow control) to the network side of the ICD to minimize the amount of control information that must flow across the cryptographic level. Control such as the ARPA Net HOST-IMP communications must be handled in such a manner, or a non-enciphered path must be provided. Moving any possible controls to the network side also reduces the amount of NCP code that must be certified, since only that portion that handles clear text has this requirement. Other possible considerations include: (1) the cost to generate and maintain separate versions of the NCP in a heterogeneous net, (2) the problems of simultaneous updates in protocols, and (3) the side-effects of different NCP implementations. Such a change would also be a step towards some level of protocol standardization which is badly needed.

The extent to which an NCP can be removed from a HOST largely depends on the manner in which the HOST "views" the net. If the net appears to be a compatible device, (e.g., a peripheral), the HOST OS can co-exist with it with minimal impact. Otherwise, the network interface will probably impact the OS in a large number of areas, which therefore means that it must reside in the HOST. A large portion of the difference is due to the existence or lack of a well defined master-slave relationship between the HOST and the net (MET-72). The ANTS system (BOU-72) has been designed as both a mini-HOST (in which it contains an NCP for its own usage) and as a "protocol front-end device" in which case it looks like a peripheral to the HOST computer, and performs all NCP functions outside of the HOST.

The current ARPA net protocols and their implementations are not adequate for secure operation since they do not deal with error control in an effective manner, and certainly were not developed with any security against malicious behavior. Therefore, there is an almost complete lack of "system integrity" considerations, which are vital to secure operation. The most desirable approach is therefore to develop a new NCP with integrity controls to handle

accidental and malicious situations, and at the same time, to implement this new NCP in a standard front-end device which would provide a major portion of the network interface. At the same time, the basic protocol selection should be reviewed to see if the message-switching protocol of Walden (WAL-72) might be better suited for a secure net. He discusses the handling of "ports" as capabilities (in an access control sense) but does not consider the potential problems of controlling the establishment of end-to-end communications paths (i.e., setting up the encipherment keys). Since the "connections" in his scheme would only exist for the flow of one message, the dialog-oriented approach that we have taken for the SC might not apply.* In contrast, the current ARPA net protocol is connection-oriented (a connection is created by control commands for use during a dialog) and therefore seems to fit well with our scheme. However, the intuitive appeal of using a message-oriented protocol for a message-switched network deserves additional attention.

The Security Controller could appear to the net as a HOST, in which case, it too would require an NCP. If this function were moved to a standard front end device, there would be no problem. If not, then the SC control program must include NCP-related functions which would thereby introduce complications in certifying it. The size increase would be considerable since typical HOST NCP's currently range in size from 10,000 to 100,000 bytes. Some subset of these functions could apply to the SC, just as the Terminal Interface Processor (TIP) of the ARPA net has implemented only a portion of the NCP (requiring somewhat less than 10,000 bytes). One possible alternative that would eliminate the need for an NCP in the SC would be to only utilize direct dial connections to the SC, regardless of the basic network architecture. In this manner, the control functions would be handled by the direct dial net while the regular network usage would be via message-switching, etc.

*The notion of connection appears to be prerequisite for end-to-end encipherment (using a separate encryption key for each dialog), and to implement the explicit opening and closing of a particular communication path. However, end-to-end protection is possible by a combination of encipherment and other protection means.

3.7.2 The SC Control Program

Since the SC is to provide a well-defined and bounded set of responses to a pre-defined set of inputs, its control program can be designed as a simple enquiry-response system. As such, it should operate as a stand-alone control program that has been developed specifically for this purpose, as opposed to attempting to utilize an existing operating system environment. The need for certification and proof of correctness further emphasizes this choice. The SC program must be as small and straight-forward as possible for these same basic reasons.

In addition to proof of correctness methods, several other design strategies should be followed. These include those of Autodin as summarized by Lipner (LIP-72):

- Use of redundant checks throughout the message processing.
- Restriction of the allowable set of user inputs.
- Segregated areas for programs, tables, and buffers.
- Table-driven to minimize program changes.*
- Two-person reviews of changes.

The following paragraphs discuss these aspects as they relate to the SC program, while addressing the aspects of its basic functional and auxiliary modules, its control issues, and the possible implementation languages that are available.

---

*The use of a table-driven scheme is probably necessary, but is in opposition to the desire to minimize the number of pointers (which make proof of correctness much more difficult).

3.7.2.1 Basic Functional Modules of the SC. The SC has several functions which it must perform in the creation and disposition of working connections, and each can be viewed as a module in the system design. These functions are:

- Responding to an initial request for service by a requestor, i.e., identifying the person/terminal, HOST computer, or other SC making the request, and retrieving the appropriate security profile for that person and/or entity.

- Authenticating the requestor by means of a password or other authenticator stored in the profile.

- Determining the specific request for access to a resource and checking where the resource is located, and being able to take the following action:

  (a) If local to the SC, check the access authorization.

  (b) If remote (at another SC), send a subset of the requestor's profile to that SC.

  (c) If receiving such a request from a remote SC, check the access authorization.

- Creating the necessary control message(s) to cause a connection to be established for an authorized request. This involves two possible conditions:

  (a) If requestor and resource are both in the same SC domain, the SC can create the connection creation messages for each.

  (b) If requestor and resource are in different SC domains, one of the SC's must create control messages for the two ends and relay one message via the other SC (as discussed in Section 3.3).

---

* The control messages would result in matching keys being inserted at the two cryptographic devices. The actual key generation would be performed in a master cryptographic device attached to the SC, but would be initiated by the SC-generated control message(s).

- "Wrapping up" a connection audit record when the SC is notified that the connection usage has been completed.

- Collect audit information at each of these steps, and either store or transmit this data based on parameterized checks built into the modules; e.g., if a person makes N incorrect attempts at providing a password.

3.7.2.2 Auxiliary SC Functional Modules. In addition to the abovementioned functions, the SC must also be able to support auxiliary needs including the following:

- Generation of one-time passwords.
- Controlled update of the profiles including passwords, access privileges, etc.

- Initiate self-checking programs that perform diagnostic or pseudo-penetration tests.

- Handle network protocol related aspects which require adherence to a set of conventions or standards for HOST-level communications.

- Handle terminal-level communications, i.e, utilize the appropriate character set and codes, insert the proper control commands, handle character echoing problems, insert the proper number of carriage return null delays, etc.

The network protocol and terminal handling functions do not have a direct impact on the secure operation of the SC control program, but do make certification more difficult due to their size, complexity, and asynchronous nature. Therefore, there may be significant advantages to removing these components to a separate machine (either real or virtual).

3.7.2.3 Control Issues. One of the primary concerns in the development of Security Controller is that of simplicity, such that it can be accredited as being secure and can be made as reliable as possible. Therefore, simplicity is a key consideration that will influence the tradeoffs in the design of the SC control program.

A second major area of influence is based on the nature of the SC usage. Requestors go through a well-defined dialog with the SC, and also have known constraints on the amount and type of information in that dialog. These factors should influence the selection of the space allocation scheme, the method of inter-module communications, the CPU scheduling mechanism, and the way in which disc I/O is handled.

A control block will be required for each active requestor, i.e., any requesting person, HOST, etc. that is currently in some phase of dialog with the SC. This block will contain sufficient space for the necessary state information and pointers, e.g., to the program module currently being utilized. (Reentrant programs are assumed for all of the SC functions.) Buffering of messages could be handled by either: (1) linking a set of one or more buffers to the control block, or (2) preassigning buffer space to each block. Due to the similar nature of all requestor-SC dialogs, there does not appear to be any need for the dynamic linking of buffers, and hence the buffer space should

be made an integral part of the control block. The factors in favor of combining the two are that all of the SC-requestor dialogs have known, similar space requirements (and hence can be pre-allocated). There is no advantage to dynamic pooling due to the fixed correspondence between the need for control blocks and the need for buffers, and because the linked list approach to handling buffer space adds complexity. There are also advantages in keeping the entire set of data together in one contiguous address space if these data are to be swapped between main memory and a disc store. One argument in favor of separating the two is that the designer may wish to have separate address space controls over the control block and buffer portions of the information. This aspect requires further investigation to determine whether it would indeed provide additional protection.

We will assume for the subsequent analysis that the control block and buffering are a contiguous address space with the fields as shown in Figure 3-7. In addition to the usual task control block information, this so-called Request Control Block (RCB) also contains a region for communication between modules, status information, an audit record, and of course, the input/output buffers.

One of the functions of the control block was to provide an indication of the current step in the requestor-SC dialog to ensure that steps were executed in the proper sequence (e.g., to provide redundant information for checking purposes) such that one could detect any possible mis-scheduling (e.g., skipping over the authorization checking step). Actual scheduling could be performed in the normal manner of having at least two queues of tasks waiting for the

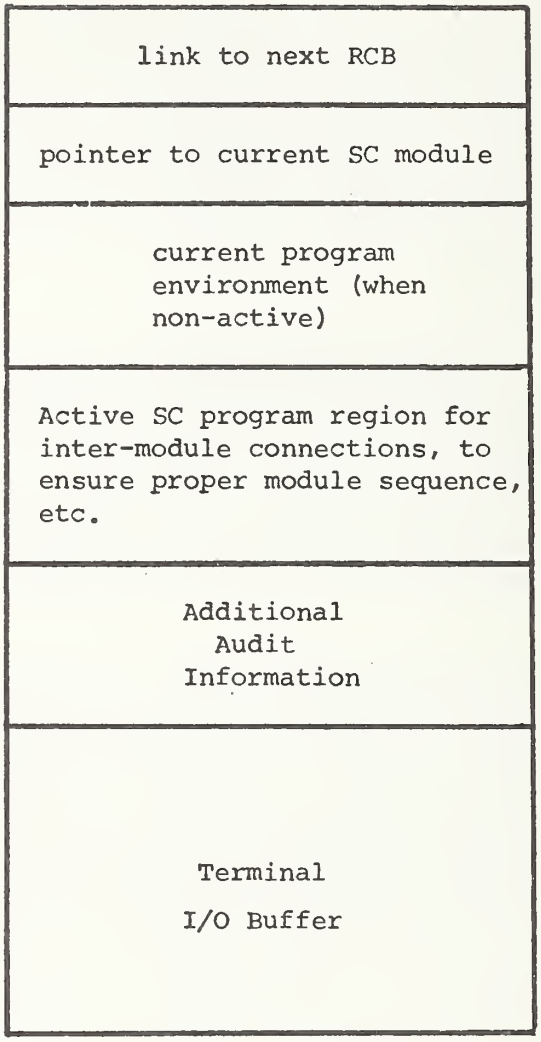


Figure 3-7. The Request Control Block (RCB) Format.

security CPU and disc respectively. We can take advantage of the inherent enquiry/response nature of the SC operation to simplify the scheduling by utilization of an approach described by Garwick (GAR-74). He proposes a self-scheduling method in which queueing is performed implicitly by the interrupt hardware, such that a given process step is run with all other process-level requests masked out. At the completion of this process step, the interrupt sources are allowed to contend for the CPU resources.[1] A "fair" priority scheme can be achieved by the appropriate order in which interrupt sources are unmasked, such that one requestor does not necessarily have an inherent priority over another.

Emergency conditions could still be handled via the interrupt mechanism, since only the process-level sources were masked, i.e., interrupts were not entirely disabled.[2] In such a case the affected requestor might have to restart his dialog since the simplified SC control program would not necessarily be able to handle an interrupt at other than its predetermined points. This is felt to be a reasonable tradeoff based on the higher level of concern that we not grant unauthorized access at the cost of occasionally requiring the requestor to retry.

I/O queueing would probably remain as an explicit function of the operating system, although this could also be handled implicitly by masking out all process-level interrupts (e.g., new requests) until both the processing and I/O have been completed. The detailed impact (and resulting simplification) of this approach would require additional investigation to determine performance aspects.

---

[1] This assumes that requests are enqueued by the hardware rather than being lost.

[2] This decision must be carefully made since it may "undo" many of the advantages of the self-scheduling design.

3.7.2.4 Program Mechanization Issues. When a program, such as that of the Security Controller, is to be certified by "proof of correctness" means, it must be developed with this in mind. Its design must proceed in a carefully controlled manner, utilizing the best available techniques to ensure consistency, simplicity, and understandability. The current technological state-of-the-art of programming would therefore dictate that structured programming techniques be applied in such design and development.

A second aspect of the design and development is that of the choice of a language, preferably a single language for expressing the design and development in a step-wise refinement manner. Book has addressed the question of what programming languages are available for use in structured programming by creating four categories of languages:*

- Those that are impossible for general usage in structured programming (e.g., FORTRAN and assembly languages)
- Those that can be utilized with some difficulty (e.g., PL/1)
- Those that are inherently structure-oriented such that it requires effort to avoid writing structures programs (e.g., PASCAL and SUE)
- Those that make it impossible not to write structured programs (there are no known examples in this class).

Any of the latter three classes could be utilized, but one might ask why not always utilize the language that enforces structured programming. The answer is in terms of the constraints and costs related with that choice. In addition

*These concepts are due to Erwin Book of SDC, but exist only as unpublished memoranda and discussions. The basic notion of such a "structured program" is that the source text representation be readable and understandable.

to the issues of loss of flexibility, efficiency, and certain real-time needs (forms of I/O handling, interrupt linkages to routines, etc.), there are also issues of (1) whether programmers are available who know the language, (2) does an appropriate compiler exist, (3) is it supported, etc. As a consequence, there is no obvious "best" language for such a development. Garwick (GAR-74) has surveyed the existing languages that might be utilized and discussed the tradeoffs involved in several including PASCAL, BLISS, MARY, and SUE. Unfortunately, none of these languages meets all of the requirements. One other alternative is that utilized by Popek at UCLA (POP-74B), namely the usage of PEESPOL, a language developed at the University of Illinois as part of the ANTS project (BOU-73). The compiler exists for the PDP-11/45 and is being maintained (at least currently). The language selection is a significant aspect that must be resolved, but currently remains an open issue.

3.7.3 Error Control/Recovery in the SC

The detection of errors is extremely important in the SC context, since such errors may result in unauthorized privileges being granted, or may result in the denial of service to legitimate users. Our fundamental objective is to be able to achieve a specified (low) probability for each of these two types of errors, with the requirement for continuity of service being inherent in the denial of service considerations. The fact that the first type of error (unauthorized access) is considered to be of greater importance than the second, leads to the notion of fail-secure operation (a term coined by Molho, MOL-70).

Denial of service can also be a significant security vulnerability in itself, and part of the SC design must include provisions for back-up in case of SC component failures. Two basic approaches are available, differing in the way that redundant equipment is to be utilized. The first is the common duplexing (or N-plexing) of equipments such that a "spare" machine exists at the site and is available to be switched in as the primary vehicle. The second basic alternative is to distribute the redundancy across two or more sites which, in the case of the SC, means that a given user would appear in two or more SC's. Thus, when the user's primary SC becomes unavailable, the user can revert to the usage of a

back-up SC. This second alternative tends to negate many of the advantages of the centralized/localized control over requestors and resources, as well as to introduce many logistic-related problems regarding updates, etc. Therefore, it will not be considered further, and we will assume that error control will be entirely by means of redundance at the individual SC's.

Failures (or errors) can be considered to be any transient or permanent deviation from some established normal operation, such that the results of a given process, or process step, differ from that which would normally be obtained for a given set of input data. We include in such failures all hardware errors, hardware-induced software errors, and any timing or data-dependent software anomalies. We specifically do not include logical or mechanical errors in the software which should be handled by certification techniques as discussed in Section 3.6. This latter form of error is due to the unsuccessful realization of the original intent of the program; information that is not available at run-time when the SC checks must be made.

The only means by which run-time failures can be tested are: (1) by redundancy checking of the process and (2) by integrity checking of the processor. Redundancy checking ranges from the usage of simple parity bits to the usage of multiple processors that execute the same processes and "vote" to determine the correct result. In contrast, integrity checking is independent of the specific processes being executed, and merely verifies (to some level of confidence) the correct operation of the system. Diagnostics and pseudo-penetration attempts are examples of this latter approach, and have already been discussed. As is usually the case, a combination of the two approaches provides a stronger system than was provided by either scheme separately. Similarly, the application of checking at several different levels provides multiple barriers (in the form of disparate checks) through which an error must find its way before it can cause any damage. The approach can be viewed as a systematic implementation of the need for redundant checks, and has been described by Grycner (GRY-74).

The systems (or level) approach to error control requires that some error checking be applied at each level in the hierarchical implementation of a given process. At the highest level, this checking would be of the final output(s) of the process, which could be checked against values obtained through either: (1) pre-assigned reasonableness bounds, (2) repeating the execution of the same process, or (3) concurrent execution of the same process on two or more processors. These can all be referred to as process level checking, but the results have greater significance as one moves from the first to the third alternatives.

Concurrent execution on multiple processors can introduce a very high degree of protection against undetected errors as long as any systematic (i.e., common to all) sources of error have been removed. These common errors can be eliminated by either providing separate (independent) operations or by time-staggering the operation of the machines such that a given transient affects them at logically different times. This might be done by inserting NOP padding at different points in the otherwise identical processes. An alternate approach would be to run different, but equivalent, processes in each processor. This approach is not theoretically feasible in general due to the difficulty (if not impossibility) of proving the equivalence of two or more programs, but may nevertheless have practical utility in selected instances.

The multiple processor alternative requires that the machines execute the same process, but synchronize at selected check points as part of determining if all (or some majority) agree. In case of disagreement, the processors may retry (i.e., to obtain either unanimous agreement or to determine the faulty processor).

The cross-checking of two or more processors could be performed by any of several approaches, but basically all share a common structure as shown in Figure 3-8, in which the process steps are checked and synchronized at several intermediate points prior to emitting any "results" (in our case, results would be connection establishment commands). Note also that the output results are error-encoded prior to the final check, at which time the protection reverts from processor redundancy to data redundancy (e.g., via error encoding such as the addition of cyclic redundancy bits).

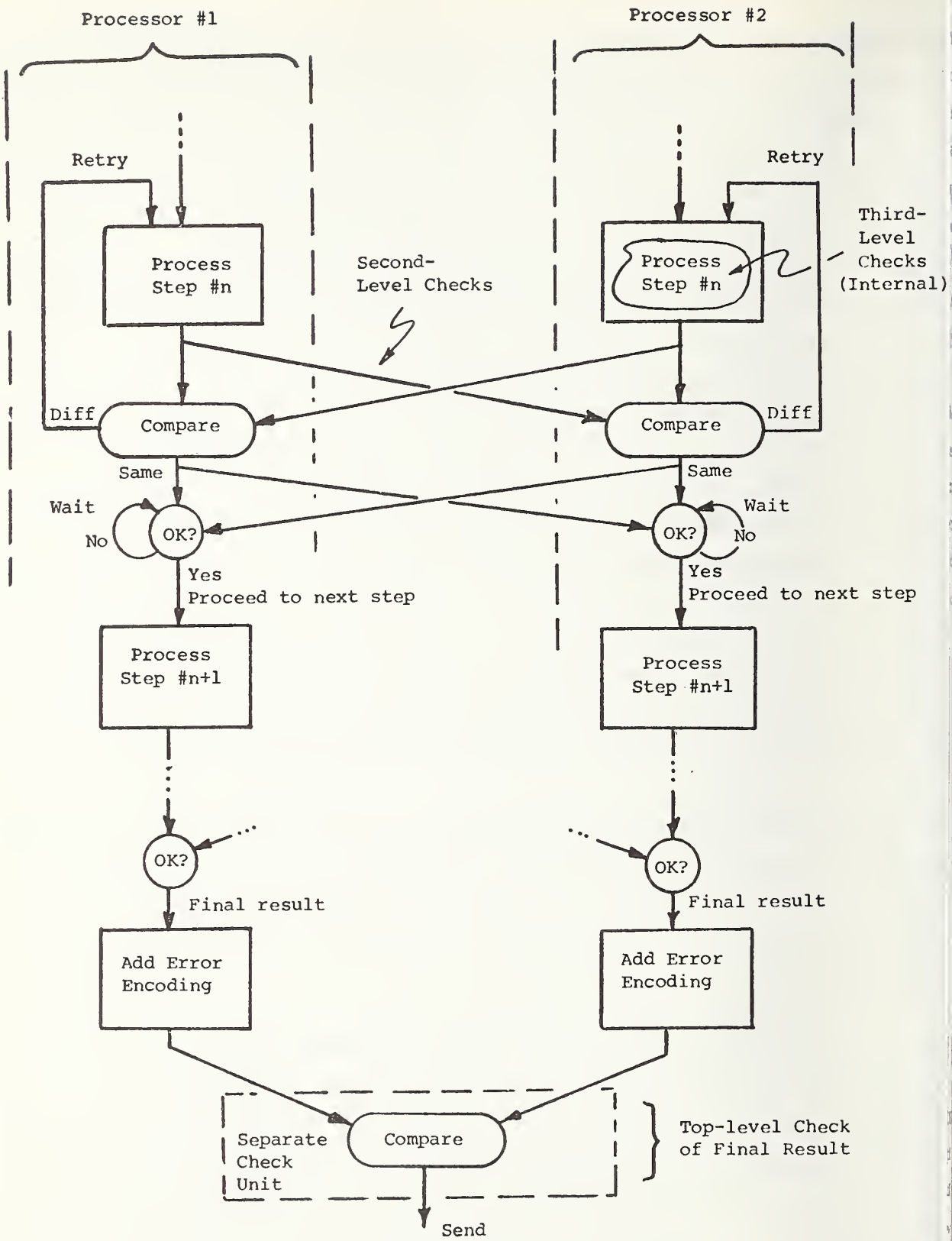


Figure 3-8. Step-Wise Cross-Checking of Two Processors

The basic cross-checking structure can be applied in any of several configurations as indicated in Figure 3-9. All three configurations revert to the same structure for the special case of only two processors, which can detect a processor failure, but can not necessarily determine which processor is wrong. For N greater than two, the system can continue operation in spite of failure(s). The optimal number of processors to ensure adequate error detection and continuity of service is a design issue that must be addressed in the context of a specific application environment and a particular candidate processor.

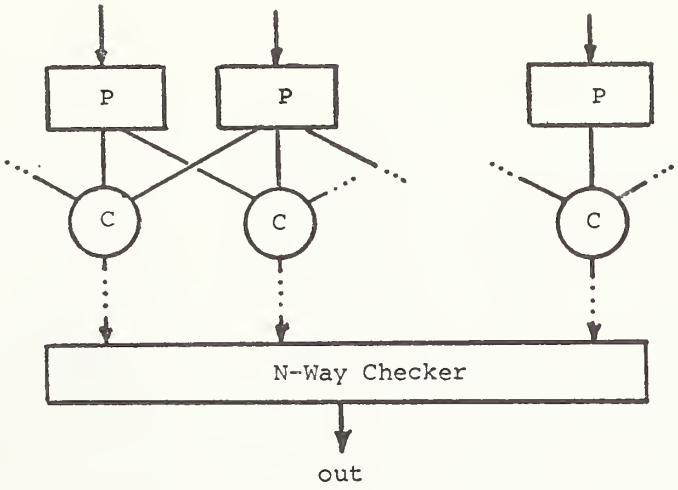
A variation of the process-level checking is to perform these tests on smaller segments (e.g., sub-processes). In an extreme case, this degenerates to checking on an instruction-by-instruction basis, which presents a very large overhead, while providing no real advantages. The granularity with which this checking should be applied can only be defined in terms of the detailed context of a particular design, but the following general guidelines can be applied.

- Checking should be performed whenever the results of an operation are to be transmitted outside of the SC system (e.g., profile data sent to a HOST).

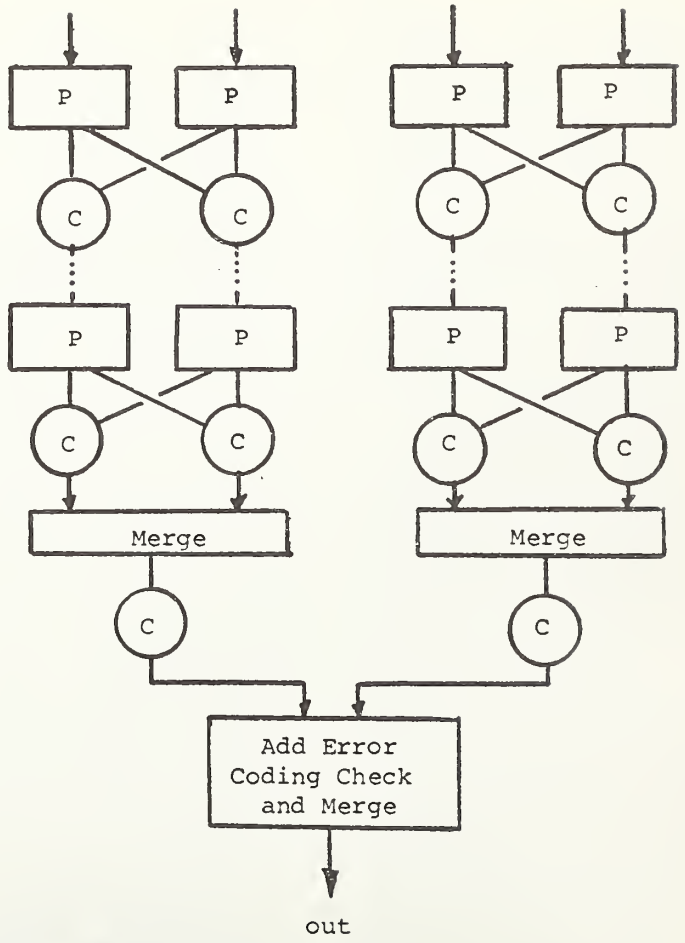
- Checking should be applied whenever a reference (read or write) is to be made to a peripheral of the SC.

- Checking should be considered whenever the expected time to rerun the process (i.e., the product of the processing time and the probability of error) exceeds that of the checking operation.

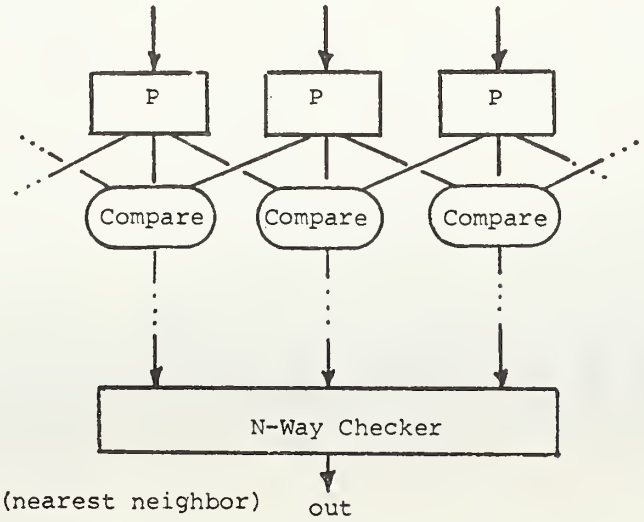
Other forms of error checks and controls should also be applied by the system to detect (if not avoid) errors, and at least to constrain damage by an error to a small, recoverable portion of the system. One example of this approach would be to require that all critical system tables be in a write protect mode unless explicitly unlocked just prior to updating selected entries, after which



(a) Fully Connected



(b) Hierarchical



(c) Circular (nearest neighbor)

Figure 3-9. Alternative Interconnections for Checking of Duplicated Processors

they would again be locked. Preferably, these tables would also be protected in a form that separately unlocks individual portions of the entries instead of via an "all or nothing" lock. Protection against errant I/O transfers must also be applied, although typical minicomputer, Direct Memory Access (DMA), controllers do not abide by the normal memory address controls. One approach would be to physically constrain the address region obtainable by the DMA; e.g., by forcing higher-order address bits to some preset values.

Another example of error control is that of check-summing critical programs (such as is done for the ARPA Net routing programs of Reference BBN-74A). A check could be made before and after the execution of a critical process if it is deemed necessary and operationally viable. For example, the checking routines that compare the outputs from the multiple processors might utilize this mode of protection.

3.7.4 SC Hardware Requirements

The hardware requirements for the SC come from: (1) the availability of adequate checking facilities, both for internal processor checks and for inter-processor checks, (2) the availability of security-related hardware features such as multiple-states of operation and address space controls, (3) its operational capabilities, and (4) the availability of appropriate software development tools (e.g., a compiler for a structured programming language).

We have discussed the language related aspects in an earlier section and will not repeat them here. The other three aspects will be discussed in the following paragraphs.

3.7.4.1 Adequate Error Control Facilities. Very few contemporary computers of the scale needed for the SC (e.g., mini-computers) have checking of any substance, with most being limited to simple parity checking of memory transfers. Registers, busses, and arithmetic/logic operations are typically not checked at all, with a few exceptions. There is some possibility that

additional checking could be obtained via the usage of a microprogrammable machine for the SC. Such checking could be in the form of micro-diagnostics, and might conceivably be able to augment run-time checking.

Checking at the individual processor level is very desirable since it provides an additional layer of protection against error-induced security violations. However, such checking can not be stated as an absolute requirement as long as the higher level checks (e.g., via multiple processors) are made. At this point, it appears that the "requirement" for internal checking should be left as a subjective design consideration.

In addition to error checking, certain other error control features might be obtained in the hardware such as Read-Only Memory (ROM) for the program regions, a watch-dog timer for detection of endless loops, and a program activated alarm to notify a security officer in case of an alarm condition.

3.7.4.2 Security-Related Hardware Facilities. The SC should have the maximum possible security hardware currently available, to protect against both accidental and malicious attempts to defeat its security control function. These facilities should include the following requirements which have been adapted from Bushkin (BUS-74):

- At least two operating states (and preferably more) to implement the least privilege concepts and to constrain the possible combinations for ease of proof of correctness.

- Control of address space to at least a "page" size block, and at least on a read/write basis (preferably including execute as a separate privilege).

- Address space controls over DMA (e.g., disc) transfers.

- A trap facility to handle any violation of the security mechanisms.

- o A register to indicate the offending instruction in any trapped violation.

- o Hardware interrupt with separately maskable interrupt sources.

- o A meaningful key-lock mechanism to disable the entry functions of the CPU's front panel.

3.7.4.3 Operational Requirements. We assume that the SC will function in an environment in which there are a large number of potential users (of the order of 2,000) while about half of these users are actively using (or requesting) computer resources. We further assume that the typical active user needs a new access (via the SC) about every 20 minutes, and will spend about one minute in the dialog with the SC to gain this access. Following the analysis of Garwick (GAR-73) the expected number of users in a dialog with the SC can be shown to be:

$$\bar{n} = N\left[{}^{t_s}/_{t_a}\right]$$

Where N is the number of active users (1000 in this example), t_s is the average service time (one minute), and t_a is the average time between arrivals (20 minutes), thereby yielding a rough estimate of 50 simultaneous users in some stage of dialog with the SC.

Disc storage must be provided for the identification, authentication, and authorization data for each possible requestor, as well as providing main storage for that subset of the requestors currently in some stage of dialog with the SC. Estimates have indicated that the disc storage requirements per requestor are:

for identification -	50 bytes
for authentication -	200
for authorization -	250
TOTAL	500 bytes

For 2,000 potential requestors, this would require about 1M bytes of disc space.

The main (e.g., core) storage requirements have been estimated to be 16,000 bytes for programs (divided evenly between the basic SC code, the I/O handlers, the terminal handlers, and the audit/status/error recovery package). The data space per requestor that is in some stage of dialog with the SC consists of the Request Control Block and its associated buffer (estimated to be 150 bytes). For 50 simultaneous requestors, the total working space must be 7500 bytes. At a maximum, this space requirement is equal to 150 bytes times the number of logical input ports which the SC will handle. A rough estimate of 10,000 bytes seems reasonable for this data storage. The total estimated program and data space is then of the order of 26,000 bytes, which is consistent with available minicomputer storage capabilities, and leaves a factor of at least two for growth potential.

Each dialog will consist of the execution of about 2,000 instructions by the SC, which at typical minicomputer speeds would require about 4 milliseconds. In addition, each dialog would require several disc accesses at from 10 to 100 msec. each, depending on the type of disc (e.g., either fixed or moving heads). The number of disc accesses will depend on the complexity of the access authorization structure (e.g., linked lists), and on the decision as to whether the active requestor RCB's are kept in main memory or are swapped from the disc. Since the RCB was assumed to include the terminal I/O buffer, we have implicitly assumed that this storage will be "core" resident for the duration of a requestor-SC dialog. For this assumption, we estimate that there will be of the order of 4 disc accesses per dialog, at an average of about 100 msec. per access (for moving head seek time plus latency). These time delays, result in an I/O queueing situation in which one request will be received per user every 15 seconds on the average, and will require an average of 0.1 seconds for the I/O service (transfer time is small compared to access time). For 50 active requestors in the SC, the expected time between I/O requests is 15 seconds ÷ 50 or 0.3 seconds. For a service time of 0.1 seconds, the queueing "traffic

intensity" is therefore 0.1/0.3 or 0.33, which leads to expected values of:

Expected number in (or waiting for) I/O service = 1/(1-0.33) = 1.5

Expected time for I/O (service plus queue delay) = $\frac{0.1}{1-0.33}$ = 0.15 sec.

Therefore, a standard moving head disc (e.g., a 2M byte cartridge) would appear to meet the operational storage requirements without producing any appreciable I/O queueing delays.

3.7.5 Performance Impact Due to Security

The added requirements of the security mechanisms have a cost in terms of not only the equipments themselves, but often in terms of some level of performance degradation. In the approach which has been described, this effect shows up in the need for an initial (pre-connection) dialog with the SC, in any subsequent HOST-level checks, and in the network overhead required to route audit information to a central site, e.g., the Network Security Center. (The impact due to cryptographic aspects will be considered separately in Section 4.)

The pre-connection dialog with the SC was estimated to require about one minute, primarily due to requestor typing delays, and is an overhead that must be paid for each working connection which, on the average, would last about twenty minutes. This effective overhead is about 5%, being more or less dependent upon the actual duration of each segment of the cycle. The time required for HOST-level checks should also be considered, but can only be estimated in the context of a given situation.

The need for audit data collection, as discussed in Section 3.5, requires that the separate pieces of a distributed audit trail be combined at some site such as a Network Security Center. We will assume that this center is one node of the network, and that audit information is sent to it via the regular network channels. For comparison purposes, it is of interest to estimate this audit collection overhead relative to the operational network involved. Three classes of usage will be considered, interactive, RJE, and file transfer.

For interactive usage, we assume that traffic is similar to that measured by Jackson and Stubbs (JAC-69) and therefore that each dialog utilizes an average of about 25 bits/sec., (a 12-character user burst and a 150-character computer burst every 75 seconds). For 1,000 active users, this results in an interactiv data traffic of 25K bps, to which we must add about 5K bps overhead, giving a total of 30K bps for the user-to-computer and computer-to-user traffic. The audit data collection for each such dialog will consist of SC-generated data of about 200 bytes and HOST-generated data that is estimated to be 500 bytes. For two-HOST usage, the total audit data for interactive use is therefore 1,200 bytes for each 20 minute connection, which for 1000 simultaneous users results in an average data rate of about 8K bps for audit information. (The audit information is about one-fourth that of the total interactive dialog.)

For RJE usage, we assume an average of one job submitted per day per user, and that the resulting output will be of the order of 200,000 bits, while the input would be only about one-tenth this volume. This adds a network traffic of 220,000 bits per job for 2,000 jobs per day, or an average requirement of about 15K bps for RJE work. If the same audit information is required (as for interactive usage) the 2,000 jobs would require 1,200 bytes each for an average audit data rate of about 1K bps.

If we also assume an average of one file transfer per day per person, with an average size of 2M bits per file, the required data rate is about 140K bps, while the audit record would again be of the order of 1K bps. The total of these requirements for 2000 potential users is then:

Type	Data Rate	Audit Rate
Interactive	30K bps	8K bps
RJE	15K bps	1K bps
File Transfer	140K bps	1K bps

The audit overhead is about 5% when averaged over the entire set of traffic, but ranges from a high of about 25% for interactive usage to a low of less than 1% for file transfers. The effect of the added audit traffic is highly dependent upon the extent to which the normal traffic loads the communication facility, since queueing and other congestion phenomena are highly non-linear. For example, the above traffic estimates would be representative of a 40-node, 40-link ARPA-like net, which for full-duplex 50K bps links and an average of 3 links per HOST-HOST connection, results in an effective capacity of about 1.4M bps. The normal traffic load would therefore be about 13% of capacity, and would be increased to 14% due to the audit loading.

4. NETWORK SECURITY AT THE INTELLIGENT CRYPTOGRAPHIC DEVICE LEVEL (ICD)

The need for secure computer networks adds several new requirements that the cryptographic devices must meet, above that of conventional point-to-point line protection. One of the most fundamental differences is the need to not only protect against unauthorized reading of messages, but also to protect against unauthorized connections between two network entities. This protection is accomplished by establishing such connections via the Security Controller, and only after the appropriate authentication and authorization tests have been passed. A second major aspect of the Intelligent Cryptographic Device (ICD) is that it must be capable of being remotely keyed, but only by the Security Controller. A third related aspect is that the protected connection must be broken when a dialog is completed, such that it can not be utilized by others in a piggy-back fashion.

The needs require that a certain amount of logic (or intelligence) be built into the control mechanism of the encryption device. The amount of such control logic will depend on a number of factors including (1) whether the cryptographic devices are multiplexed or dedicated, (2) whether there is a Master-Slave relationship in the inter-crypto device control, and (3) whether the devices need to be able to relay connection creation messages along to another ICD. Each of these factors will be considered in subsequent sections of this report.

Before proceeding with the discussion of the network ICD, we will briefly describe the point-to-point variety of crypto device. The basic portion of a typical encipherment/decipherment scheme is shown in Figure 4-1, in which a matching key has been inserted at each end of a communications path. The resulting bit stream from the generator at each end is the same, and since the exclusive-OR is self-inverting, the clear text is retrieved at the receiving end. AN ICD must include other features as well, with a minimal set of requirements for an ICD being shown in Figure 4-2 which is intended as an ultra-simplified conceptual model of how the remote keying, etc., would be handled. The added features for the ICD include a key select mechanism, which can select either a null key (no encipherment), a private key (known only by the Master ICD at the SC), or the working key which would be set up specifically

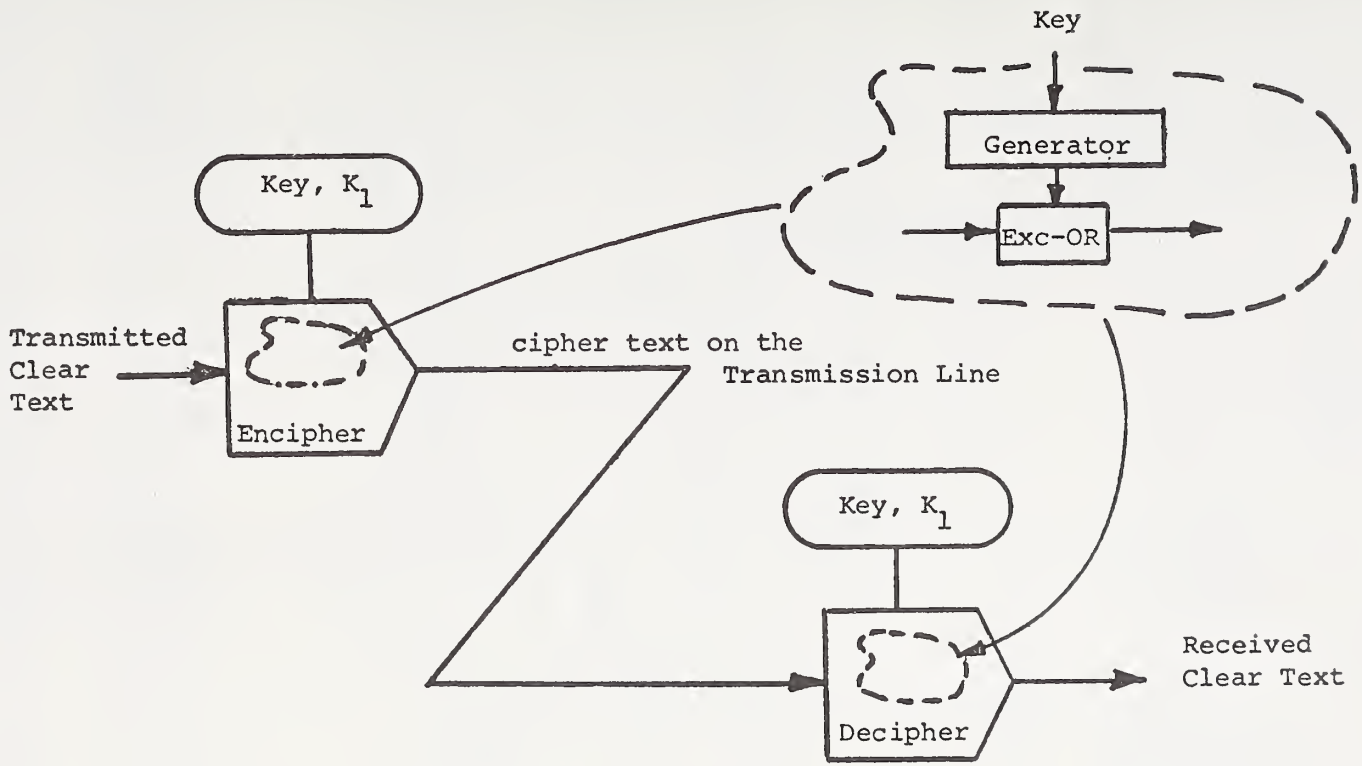


Figure 4-1. An Exclusive-OR Encipherment/Decipherment Scheme

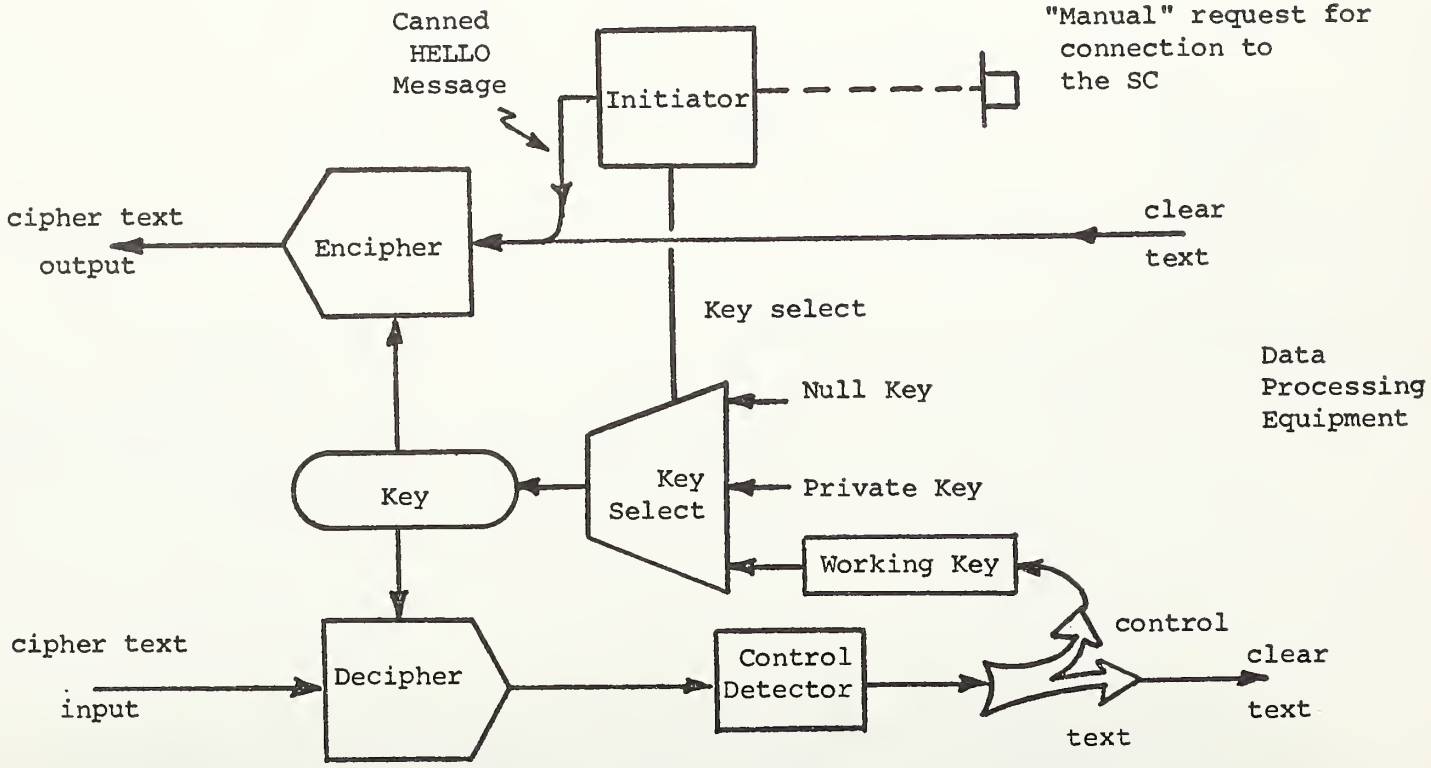


Figure 4-2. Simplified Conceptual Model of an Intelligent Cryptographic Device

for the particular dialog. The null key would only be utilized for sending a clear "HELLO/id" message that identifies the requesting ICD, and only as a result of the "request for connection" signal being applied (a manual button depression in this figure). The private key would only be utilized for distributing working keys (i.e., to decipher a working key as received from the Master-ICD at the SC). This distribution would be via embedded control commands detected by the "Control Detector" box, and therefore shunted to the working key register instead of being passed on to the data processing equipment as clear text information.

Other needs in the ICD will become apparent as we proceed, but this simple model is adequate for our initial discussions.

4.1 THE ICD IDENTIFICATION/AUTHENTICATION MECHANISMS

As mentioned in the introduction, the canned "HELLO/id" message would include identification information, e.g., "HELLO, THIS IS #412." Since this message would be sent in the clear, it could easily be read and/or forged. However, this does not really matter since it is merely an identifier, and does not authenticate the ICD in any way.[1] Instead, the Master ICD at the SC would then look up the unique private key of this particular ICD, and utilize it to send a temporary working key to the requesting ICD. The fact that the two ends can subsequently communicate in the temporary working key is implicit authentication, since only these two entities would know the private key.[2]

---

[1] The design must preclude the threat in which a forged hello/id message is sent to the SC such that the SC will re-key an ICD that is in use, thereby destroying an authorized connection. A special initialization mode should be adequate protection.

[2] Katzan (KAT-73) summarized a similar security handshake scheme that he attributes to Feistel, Notz, and Smith of IBM.

A special test message "handshake" might be utilized as a quick test that such communication exists. (Note that the test message must be more than a simple "echo-back" since that would work in any case. Any deterministic response is adequate.)

Authentication of the ICD is important for several reasons, including (1) the assurance that it provides that one end is not spoofing the other, and (2) the implicit authentication that it provides to its attached data processing equipment (as long as appropriate physical and procedural controls are maintained). It might also provide a mechanism by which a given device could have two or more roles, e.g., could operate at either Secret or Top Secret levels at different times during the day. This would require that either two or more identifiers and private keys be "wired in" to the ICD or that the effective id/private key be the composite of that of the ICD and values provided by the data processing equipment. Either case would require an extension to our simple model of Figure 4-2.

One could extend the composite id/private key notion to include the person utilizing the data processing equipment, but there does not appear to be any advantage to such an arrangement.

The strength of the protection provided by the ICD's is completely dependent upon the secrecy of the private keys, but since they are only to be utilized for enciphering working keys, there is limited vulnerability to their repeated usage. (The working keys are essentially random numbers.) Therefore, their replacement at selected intervals would be primarily to limit the duration of exposure if a key were compromised. Update would be via some non-network means such as manual distribution and insertion.

4.2 THE ICD ACCESS REQUEST/AUTHORIZATION MECHANISMS

The primary authorization function of the ICD is its role in effectively blocking communication between two entities until that communication has been authorized and established by the SC. As such, it serves an enforcement role that requires all requestors to go through the SC, thereby ensuring that it is always invoked.

A second area which can be controlled by the ICD is that of handling devices that dynamically change their security level. This control is an extension of the idea of providing an ICD with a set of two or more identifiers and authenticators, such that the device can serve a multiple role. For example, a HOST computer could operate at the Secret level most of the day with its ICD having one identifier/authenticator pair, but could change to Top Secret for a period of time, during which its identifier/authenticator pair would be modified by either selecting a second set of values or by forming a composite with HOST-provided data.

There is also an inherent authorization issue involved in the master/slave control structure of the ICD's, with the SC having the Master-ICD and the various requestor/resource entities having slave devices. This distinction is, of course, due to the special needs of the SC, and the desire to clearly separate the issues related to ICD's from those at the user/HOST level. The Master-ICD would be the only device that would "know" the private keys of the ICD's in that particular SC domain, as well as the pair-wise keys for setting up SC-to-SC communications.

4.3 ACCESS CONTROL AT THE ICD LEVEL; ESTABLISHMENT OF CONNECTIONS

As mentioned earlier, the control over the creation of connections is basically by means of the controlled establishment of working keys by the SC. It can enforce this control since only the ICD of the SC is (1) initialized to understand the clear HELLO/id message, and (2) able to establish working keys via the private key mechanism.

Since all communications between the SC and the other entities will be by means of bit-serial (half-or full-duplex) paths, the control and data information must share the same channels, just as is typically the case for the various line disciplines which mix control and data in the same stream. We can therefore assume that the key distribution scheme must be based on the usage of embedded control commands, which must be recognized and acted upon at the appropriate time and place.

A connection is created when identical working keys have been inserted at the two end ICD's, and as discussed in Section 3.3, there are several options as to how this distribution might actually be handled. In one case, the SC/Master-ICD would send the keys to the two ends by means of separate control messages, thereby "priming" the two ICD's so that they can communicate. The other alternatives were variations of a relay scheme in which the Master-ICD would send both copies of the working key to one ICD, which would then remove its copy (and related commands) and relay the rest on to the second ICD.

This approach to key distribution requires that an ICD not only be able to relay messages, but also be able to extract and execute control commands that are "addressed" to it. Several issues need to be considered in this regard, including:

- What control primitives should be provided for the ICD's.
- How should the embedded control commands be addressed to the appropriate ICD that is to act upon them (since a relayed connection-creation message would involve at least three ICD's).
- How can one ensure that the private key of the second ICD is not "exposed" during the relay process (i.e., whether the relaying ICD could gain any information regarding the other's private key).
- How should these primitives be formed into appropriate control strings to set up the matching keys via the use of the private keys.
- Should commands be executed "on the fly" or only after error-checking.
- How should the Master-ICD be notified of the status of the connection establishment.
- How should connections be terminated (normal and error conditions).

Each of these issues will be discussed in more detail in the following paragraphs.

4.3.1 Control Primitives

The basic issues involved in selecting the ICD control primitives are that they (1) form a sufficient set to meet the known requirements, (2) are open-ended (i.e., capable of being extended as new requirements become known), and (3) are simple to understand and implement. Three variations of the control primitives were developed within these guidelines, and lead us to the conclusion that, at a minimum, two primitives were required; one to insert a new working key and the second to "skip over" a particular text string without scanning it for control commands (transparent text). A third primitive-like consideration was that of how and when the private key should be "commanded." In two of the designs, this was handled explicitly; either by a "Set Private Key" command or by a special flag in the "Insert New Key" command to indicate that the key is really to be the private key. In the other design, the private key was inserted automatically after having sent out the "HELLO/id" message, with the implicit understanding that the next message to be sent to it would be a working key (enciphered in the private key). (In multiplexed ICD's, these commands and "modes" should be considered on a per channel basis.) Other control primitives were added in certain cases as the designers considered a larger context of the networking environment (e.g., the need for message leaders and primary/secondary keys for enciphering the message text and leaders differently). We will discuss these auxiliary commands after considering some variations of the two basic commands.

4.3.1.1 Insertion of Working Keys. Our various test designs considered several ways that the new working key would be inserted; differing primarily in whether the insertion would be on input and/or output (e.g., of a Store-and-forward Relay message). One design allowed the choice to be made explicitly via a field in the command string. A second restricted key changes to output, and the third allowed changes only on input. Having all three allows maximum flexibility, but at the cost of having a much larger set of potentially valid sequences that might result in maliciously or accidentally induced security problems. For example, it would be possible for someone to play the role of the SC and spoof a requestor under the circumstances in which the "HELLO/id" message is sent in the clear and the SC is to respond by sending "Set Private Key" in the clear followed by "Insert Working Key"

enciphered in the private key. The spoofer would merely send the "Insert Working Key" command in the clear and all of the implicit authentication would apparently be met since the two parties could communicate. The basic flaw was in allowing the external (SC-like) device to issue the "Set Private Key" command. The other two designs constrained this function, one with an explicit, but internal, command, and the other performing it implicitly.

4.3.1.2 Handling Transparent Text. Each design required some way to indicate that a given sequence of text was not to be scanned for embedded control commands, either because it was to be interpreted at some other point, or because it was really text (but might contain the bit patterns of the commands). One design utilized "brackets" around such text strings in a manner similar to that in which DLE STX and DLE ETX ASCII character sequences enclose transparent text. Since the control string may accidentally occur inside the sequence, it is necessary to scan for it and duplicate it on transmission and then discard one of any such pairs upon reception (just as done in the DLE case). The other two designs utilized explicit length fields for the transparent text, with a command of the form:

Enter Transparent Mode <length>

In the ICD context, the choice between the two approaches is a design issue that affects the type of circuitry (e.g., counters for length fields versus gates for pattern detection) and the buffering requirements (i.e., the character-doubling can result in message lengths of up to twice their intended length).

A third approach, discarded as being too inflexible, is using a predefined, rigid format for the control messages so that control and data fields are to be found in known places within the message. While possibly adequate in a well defined, static environment, this approach is not appropriate for a new development such as the ICD/SC.

4.3.1.3 Auxiliary Commands. If messages are to have leaders specifying source, destination, priority, classification level, etc., it may be necessary to restrict the formulation of such leader information to the SC, and have the leader set up as part of the connection establishment. This would require a separate control command such as:

Set leader <leader content>.

One other aspect of message leaders is whether they should appear as clear information or if they should be enciphered. If they are left in the clear, some traffic analysis information may be gained by a wire-tapper, or potentially some of the leader information might be changed maliciously (e.g., its security level). If leaders are to be enciphered, they should utilize a different key from that of the message text which should use a key uniquely assigned to that given communication. To meet this need, one of the designs utilized two working keys; a primary key for the message text and a secondary key for the leader. It therefore required a control command to switch back-and-forth between these two keys, and selected one of the form,

Revert to Secondary Key <length>

in which the key was changed for the particular field length, and the text was treated in a transparent manner.

4.3.2 Addressing of Embedded Control Commands

Different portions of the control command strings should be executed by different ICD's along a relay path, such that a means is required for "addressing" each command to the appropriate ICD. This can be done by either explicitly labeling the commands with the name of the ICD which is to execute it, or implicitly by having each consecutive ICD execute all of the commands that it "sees" and "hiding" all others from its view. There are two implications of this latter scheme; (1) commands should be deleted after being executed and (2) selected portions of the message should be skipped over without scanning them for control commands. This is basically the notion of transparent text which was discussed in the previous section.

One of the test designs utilized the explicit approach to addressing with command structure of the form:

<command><address><key>

in which the device to execute each command is labeled, and therefore the "used" command can be either dropped or carried along depending on which is more convenient. The other two designs utilized the implicit addressing schemes and either NOP'ed or deleted the command from the string. The latter also compresses the control string, but requires buffering. The implicit command addressing utilizes the address fields of the composite message leader, or nested leaders, (as discussed in Section 3.3.4) to identify which ICD executes which portion of the embedded commands.

4.3.3 Control Strings Using the Primitives

As previously stated, the control commands (primitives) will be embedded in text strings, which will then cause the appropriate connections to be established via the interpretation of these strings at each ICD along the path (typically in a relay fashion). We have found that these control strings look very complex, but upon closer inspection, one finds that they conform to a straightforward and static set of macro-like templates which the SC and its ICD could readily utilize to create the connections.

4.3.4 Concern for Errors in Control Commands

Errors encountered in a control message must be handled with special care since the ICDs at the two ends may be left in different states, and hence may have to reinitialize via the SC. This condition would be similar to that encountered by loss of crypto-sync in a running key scheme; a condition that could otherwise be avoided by using a self-synchronizing method of cryptographic protection (see Section 4.4.1).

If a sending ICD has a message buffer, it can hold a copy of a control message until it has successfully reached the recipient, (and retransmit if necessary).

Similarly, if a receiving ICD has a buffer it can delay executing any commands until the entire control message has been received and error checked. This would require that our conceptual model of an ICD be modified from that of Figure 4-2 to include a storage buffer between the deciphering unit and the control detector. Once such a buffer is included in the design, it can begin to influence many other aspects of the ICD usage. We mentioned earlier that a buffer would be needed if "used" control commands were to be removed (as opposed to being NOP'ed). Similarly, other buffer related advantages could be found such as the inherent forming of messages into buffer-size packets for transmission, the ability to cancel a line that has been input, but not sent, etc. This would imply that there should be an output buffer as well as an input buffer (or at least that one buffer could serve both purposes). There would be significant advantages in making the ICD respond in a full-duplex mode, thereby needing separate input and output buffers. If an ICD is to buffer a control message for possible retransmission, it must of course retain a copy of the message with the embedded control commands (i.e., not removed or NOP'ed). This means that the information would have to be stored in the input buffer, thereby restricting full-duplex operation during relay messages. (This does not appear to be a problem, but should be considered in the detailed design.)

4.3.5 Notifying the Master ICD of the Connection Status

As our investigation has proceeded, it has become increasingly clear that the SC needs some feedback regarding whether or not a connection was successfully established. For the relay-mode of connection creation, this function can easily be handled by merely adding one more relay operation; namely from the destination ICD back to the Master-ICD at the SC. This would complete the loop and thereby inform the SC of the connection status. It would require that the destination ICD "fill in the blanks," e.g., insert a condition-code, to indicate the status, and would require that the SC set up one additional encipherment key to be utilized for this ICD-to-SC (Master-ICD) transmission.

For the non relay mode, in which the Master-ICD would "prime" the two end ICD's, a different approach would be necessary. Either one of the ICD's could be charged with performing this function (or both) in a similar manner to that required to notify the SC when a connection is broken. However, the two function are quite different when an ICD has a single port to work with, since in the former case it is "tied up" with the working connection, while it is free after having broken a connection.

4.4 ACCESS CONTROL AT THE ICD LEVEL; USAGE OF A CONNECTION

In the simplest case, two ICD's might be considered to be in-line with the communications and therefore would be transparent to the usage of the line, just as a pair of modems perform their transformation/inverse transformation, and otherwise are transparent to the communication. However, the cryptographic equipments do introduce side effects, primarily related to their synchronization requirement Therefore, we should consider the various options that we have in this area, and determine their network-related tradeoffs. We also need to consider the impact of utilizing multiplexed cryptographic equipment, particularly at multi-port devices such as HOST computers, and the associated problems that multiplexing introduces due to the need to pass more control information past the encryption mechanism. Finally, we need to consider the effects of errors on all of these factors and the performance degradation due to the usage of the ICD's. Each of these topics will be treated in the following paragraphs.

4.4.1 Encipherment Scheme Considerations

There are two basic encipherment schemes: (1) block-by-block substitution, and (2) stream encipherment by exclusive-ORing clear text characters with pseudo-random character streams. The first scheme is called Electronic Code Book (ECB) mode and the second is called Cipher Feedback (CFB) mode. Both modes require synchronization. The block mode requires that the transmitter and receiver each recognize blocks. The CFB mode requires that the enciphering transmitter and the deciphering receiver must each utilize the identical pseudo-random stream.

The Data Encryption Standard algorithm can be used in either of these modes. Messages should be synchronized separately when the CFB mode is used. An initializing vector must be transmitted as a preamble of each message in this mode. ECB mode requires no initialization.

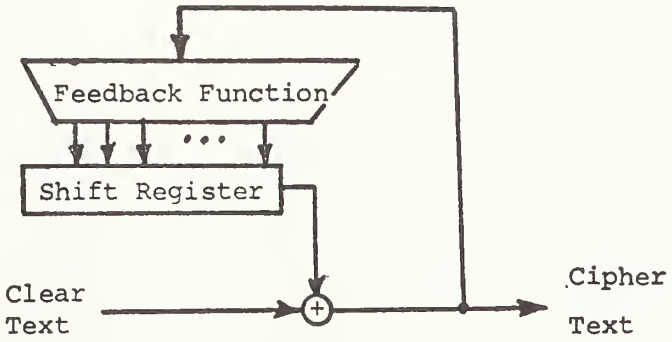
The classical bit-stream additive approach to encipherment has disadvantages making it unattractive for network usage. These problems stem from its inherent need for synchronization of the two random number generators, a problem that has been recognized and overcome for point-to-point operation, but which magnifies considerably in network environment which may require re-ordering of messages due to priority needs, sophisticated line disciplines such as Go-Back-N, etc.

Since synchronization problems are the basis of these difficulties, it is desirable to look for schemes which are inherently self-synchronizing. One such technique is to utilize the transmitted cipher text for synchronization since it is available at both the sending and receiving ends. Such schemes are well known and have been described in a number of papers in the open literature.** Each author describes a variation of the same basic theme as shown in Figure 4-3. Connection of a pair of such devices is shown in Figure 4-4, which indicates the operation of the cipher text feedback into the shift registers. When each register has the same content (the "random" sequence, XRPQ, in this example), and has the same function, they will both generate the same sequence of pseudo-random bits for enciphering/deciphering. Since the register content is available to any eaves dropper (e.g., it is the cipher text itself), the secrecy must reside in the function, which is our key in this case.

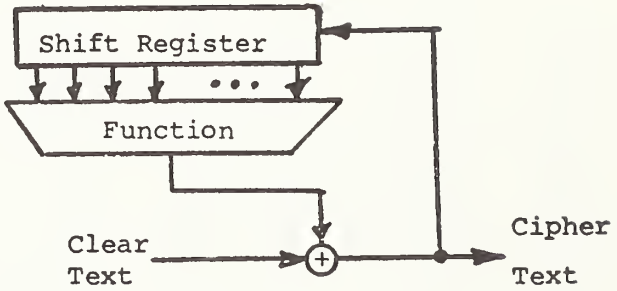
---

*The exclusive-OR operation is self-inverting since $b \oplus b = 0$ for $b = 0, 1$.

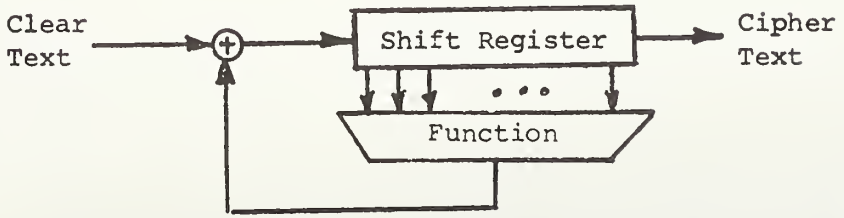
**Savage, J. E., "Some Simple Self-Synchronizing Data Scramblers," Bell System Tech. Jo., Feb. 1967, pp 449-487.
Torrieri, D. J., "Word Error Rates in Cryptographic Ensembles," NRL Report 7616, Oct. 1973.
Golumb, S. W., "Shift Register Sequences," Holden-Day, Inc., 1967



(a) Torrieri's model (NRL)



(b) Savage's (Bell Sys.) self-sync. data scrambler



(c) Golomb's model

Figure 4-3. Variations of Self-Synchronizing Schemes Using Cipher Text Feedback

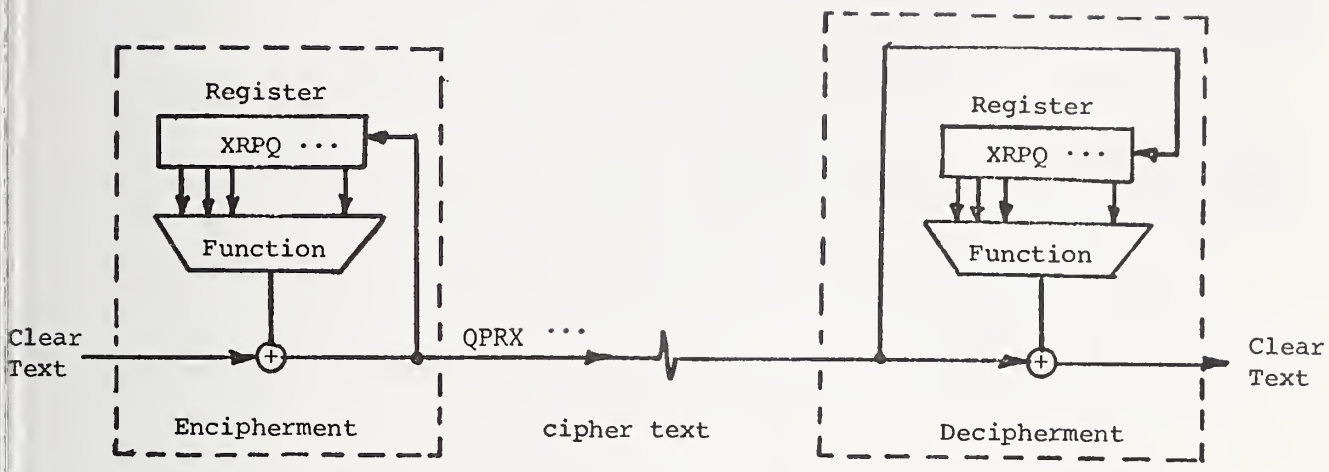


Figure 4-4. A Pair of Encipherment/Decipherment Devices

The self-synchronizing scheme has some disadvantages as well, the most commonly discussed being its error propagation problem. Any one-bit error introduced in the cipher text stream will continue to affect the decipherment process until it is completely shifted out of the register, i.e., if the register is N-bits in length, then any error will result in N-bits being garbled. This is a problem for applications which utilize English text which might be understandable with occasional 1-bit errors, but not with bursts of N-bits in error. For computer-oriented transmissions, all errors are typically treated in the same manner, so this error propagation is not of concern unless it extends beyond a message boundary and thereby causes a second message to also be lost.

Another potential disadvantage of the self-synchronizing scheme is the need to transmit an N-bit prelude to establish synchronization, i.e., to ensure that the two N-bit shift registers contain the same values. This is necessary whenever a key change is made, and is a definite consideration for multiplexed

crypto devices in which the key may be changed for each message handled. (A possible solution to eliminate this overhead is to insert some "deterministic" bit pattern into the shift-register at each key change.)

These disadvantages are quite minor compared to the advantages of the self-synchronizing scheme for network usage. These advantages (over the pseudo-random sequence generators) include:

- Minimal concern for loss of synchronization (much easier to re-establish).
- Don't have to store previous initializing vectors (IV's) in case of error (N IV's for Go-Back-N protocols).
- Can decipher messages in a different order than they were enciphered; e.g., to be able to handle priority messages that got ahead of regular message in going through the net, or to allow reassembly of message packets inside a HOST.

The advantages of self-synchronizing schemes are so great that we will only consider them in subsequent analysis.

4.4.2 Crypto-Multiplexing Considerations

Multiplexed cryptographic equipment is desirable from an economic point of view, and their development has been recommended by Anderson (AND-72), and other members of the ESD Security Panel. Their reasons include:

- Minimized costs, operator controls, space and other environmental requirements.
- Provide more than one secure communications path via the same transmission link, primarily on a time multiplexed basis.

They projected that a prototype model could be available in FY 76, and that the device could also be designed to provide authentication (similar to the way that the ICD would authenticate the device to which it is attached).

Multiplexing of cryptographic devices is a natural extension of the ICD concepts which is consistent with our earlier developments in terms of control, buffering, etc. A message leader would be received and would thereby indicate the key to be utilized via its source and destination. Based on this information, the ICD would retrieve the appropriate key and decipher the message.

Multiplexing could also be on the basis of packets (pieces of messages), characters, or even bits, although the overhead of switching keys would be increasingly large as one moved toward the finer level of multiplexing. Therefore, message level multiplexing is felt to be the optimal choice, particularly if a synchronizing prelude is required for each block of data (i.e., self-synchronizing scheme).

4.4.3 Control/Data Considerations

Certain control information must be passed between the HOST-level and communications-level interfaces without being "randomized" by the crypto function as indicated in Figure 4-5. This information includes:

- Timing information, e.g., to indicate the beginning and end of a message.
- Message "type" information.
- Source, destination (or at least an identifier for a particular source-destination pair).
- HOST-Network status information.

The design of this control path must emphasize simplicity and understandability to ensure that it can not be utilized in any way to circumvent the cryptographic function, either accidentally or maliciously.

One additional concern for separating control and data information occurs when encipherment is desired at both a mini-HOST system and at a terminal that is connected to it. This situation would occur when end-to-end encryption requires encryption capability at the terminal only while other applications, such as remote batch job entry, require encryption at the mini-HOST itself.

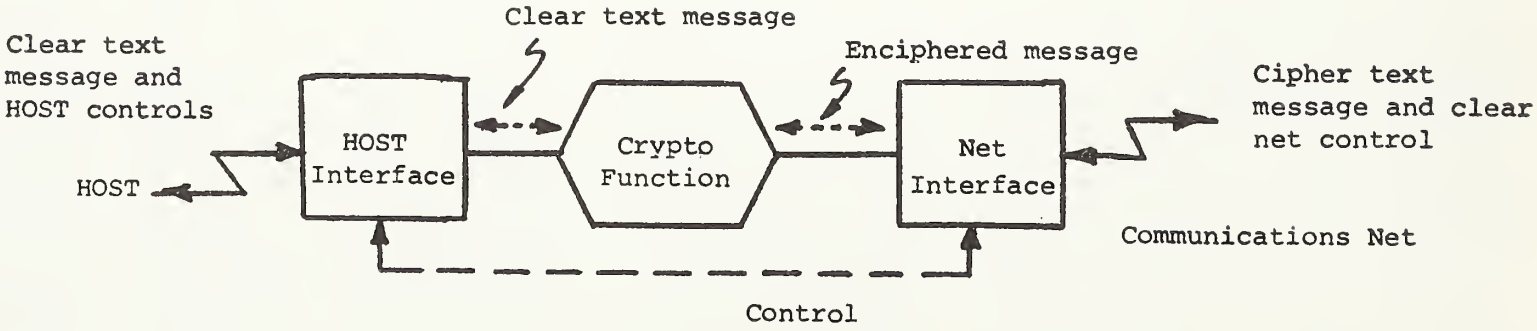


Figure 4-5. A Simplified Cryptographic Device and Its Equipment

The terminal must be able to communicate control information to the mini-HOST, but the actual terminal data traffic should be undecipherable by it. The terminal ICD must therefore be able to change its keying for control information, either to a null key or to a key that is known by the mini-HOST ICD. The latter scheme would also require that the mini-HOST and its ICD be configured such that information can flow back to the mini-HOST after being sent through the ICD.

One solution to this need is via the use of a composite encipherment scheme such that the transformations made by terminal and mini-HOST ICD's combine to be the same as that at the receiving ICD. The fact that the mini-HOST must scan the enciphered terminal character string for control commands means that some means is also required for handling the binary text problem (i.e., the accidental occurrence of control bit patterns in arbitrary bit sequences).

4.4.4 Error Control

It appears, for a number of reasons, that each ICD should have the capability of buffering a message, and thereby holding a copy of it until it is acknowledged as being successfully received. In combination with the self-synchronizing crypto scheme, this would provide an adequate error recovery scheme at the ICD level.

In order to recover from an error, one must first detect that the error has occurred. This detection is typically handled by appending check bits to each block which is transmitted, and testing upon receipt to see if an error has occurred (e.g., checking for a null remainder in a cyclic "division" check). The type of check should be selected based on the type of errors expected, (e.g., burst, single-bit, or permuted characters), since the effectiveness of each method is highly dependent upon the error source. This is one of the reasons why error checking should be performed separately at each level. (There are also throughput and response time considerations.)

Error checking at the ICD-level involves one additional decision, namely whether checking should be performed on the clear or cipher text. Checking of the clear text is desirable since it would also detect any errors in encipherment or decipherment, but such checks would not detect errors related to the communications net and the interface between it and the ICD, e.g., in the leaders. Therefore, a multi-level scheme seems best as shown in Figure 4-6. The higher level (clear text message) check might be made in the HOST computer (software) or at the last possible point in the ICD-to-device interface. Putting the check inside the HOST allows one to check the interface and the HOST-level handler, but there is no comparable mechanism for checking in a non-programmable device (e.g., a terminal). The check could be made in software for HOST's and in the ICD for terminals, but this arrangement must be approached with considerable care due to its "cross-coupling" across protocol levels. Additional effort is also required to determine the expected type of errors that would be induced by the ICD and its interface to the data processing equipment (e.g., the burst error effect of self-sync error propagation).

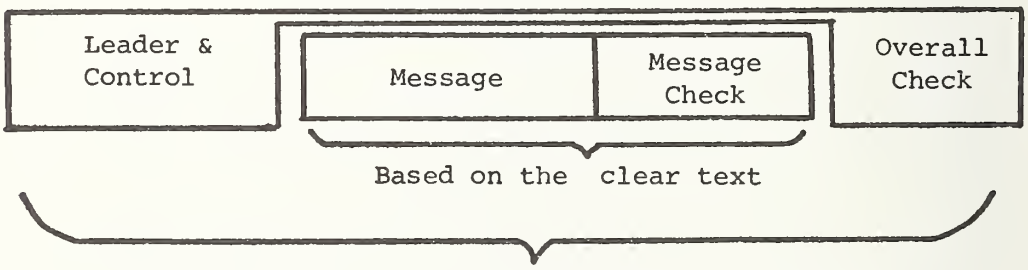


Figure 4-6. Two-Level Error Checking

In addition to errors encountered in messages, error conditions can result in the loss of an expected message or in the arrival of an unexpected message. The former would be handled by a time-out (probably at a higher protocol level) while the latter would typically be discarded. Due to the encipherment, the arrival of an unexpected message would result in jibberish unless it was a deliberate replay of a recorded legitimate message, in which case it would be detected only if message sequence sequence numbers were utilized. Message sequence numbers introduce a new source of synchronization problems whenever the two ends get out of step on the sequence numbers. However, when this happens, it does indicate the loss of a message, and thereby serves as an additional form of error control.

Baran (BAR-64) suggested a sequencing scheme which he called a "pre-filtering key." Instead of using sequential integers, he utilized 20-bit pseudo-random numbers. Katzan (KAT-73) discusses a similar method which includes one password of a predetermined sequence with each message. Postel (POS-74) has addresssed a larger scope of problems in considering sequence numbers in the context of HOST-to-HOST protocol, and has considered deadlocks and critical race conditions related to their usage. These effects demonstrate the side-effects which may result from the introduction of an apparently "harmless" idea such as adding sequence numbers to messages.

4.4.5 Breaking a Connection

When two entities have completed their usage of an SC-established connection, that connection must be broken to avoid further unauthorized usage in a piggy-back fashion, and the SC should also be notified to complete its audit record for that connection. One problem immediately arises; namely, how do we know when the usage of a connection is completed. Determining this condition may require action by one or both parties such as energizing a special control line, sending a control character, etc., or it may be built-in to the HOST and/or network interfaces (e.g., utilizing the carrier detect signal in a direct dial network). The indication may be complicated considerably by multiplexing of the ICD, and special attention must be given to this aspect in the detailed

design. An additional concern is being able to handle "dead HOST" conditions; i.e., a watch-dog timer or other active-sensing mechanism may be desirable to sense such conditions and automatically abort any connections.

Once the ICD has been notified that a connection is to be broken, one or both of the ICD's should notify the master ICD at the SC. This operation could be via a variation of the "HELLO/id" message, although any such clear-text message is subject to spoofing.

Protection could be added by requiring that the ICD first establish an enciphered connection to the SC and then send a canned message stating that its previous working connection has been broken. Both ICD's should perform this operation to handle special cases such as terminal-to-terminal and inter-domain connections.

After notifying the SC that the connection is broken, the ICD must be reinitialized, i.e., set to receive in its private key. This operation could be a built-in feature of the ICD or could be handled by an explicit control message from the SC. The former is preferable since we would not necessarily need a "Set Private Key" command other than for this operation.

4.4.6 Performance Impact Due to Security

The addition of security as a network requirement affects performance by both positive and negative factors, depending on the metric involved. The increased concern for system integrity results in improved error detection and more complete exception handling controls, thereby providing a positive impact on overall performance. However, if the metric of concern is usage of network "bandwidth," a degradation is seen due to the need for more extensive checking, spoofing protection, encryption keying, and encryption control data (such as the 64-bit prelude required in CFB operation). However, designing solely for efficient usage of communication capacity results in local component level optimization. Recent value-added networks, such as the ARPA network, have shown that utilization of telecommunication links is not necessarily a good figure of

merit for a network. User-level performance figures, such as response time and communication quality, are often much more important. The effect of security is to contribute to the values added to the network, at some degradation to the communication efficiency.

A quantitative evaluation of the overhead due to encryption security can only be made in the specific context of a given protection approach. The following factors must be considered in the determination of this overhead. No one system would utilize all of these approaches, so the contents of the list should be regarded as examples rather than as a suggested set of controls.

1. If clear text data blocks are less than 64 bits in length, ECB operation will result in a corresponding inefficiency in transmission usage.

2. If error control bits are included in the 64-bit block as a non-forgeable check-sum, this overhead must be included as above.

3. Sequence numbers or time-of-day information may be embedded in each 64-bit block to detect spoofing threats such as the recording and subsequent playback of certain messages (e.g., a funds transfer authorization message).

4. A portion of the previously received message may be embedded in each outgoing message to acknowledge the previous message and to provide a "security handshake" (to authenticate that the receiver of the messages can indeed decipher the messages).

5. Key distribution is via the same communication facility as regular messages and therefore results in some small loss of transmission throughput.

6. If CFB operation is used, there is a 64-bit prelude for each separately synchronized message or packet.

A particular network design will utilize some subset of the above approaches in providing its security mechanisms. The subset should be evaluated to determine the overhead for that particular set of controls, and alternative designs should be considered to determine the optimal protection mechanisms for a given network and set of threat conditions.

4.5 SECURITY MONITORING BY THE ICD

The ICD has a very limited context of its overall usage, and therefore can only augment other monitoring functions and provide some degree of protection against accidentally induced security flaws. This latter category includes self-monitoring of its own operation, as well as some level checking against improper usage of a connection. Each of these areas will be briefly discussed in the following paragraphs.

4.5.1. Self-Monitoring of the ICD Operation

The ICD should be designed in a manner that will detect with a high degree of confidence any operational errors such as improper encipherment (in particular, the sending of clear text when it was intended to be enciphered), and an imbalance in the randomization; at least to the level of 1/0 statistics. Other tests are probably available, but are outside the scope of this investigation.

4.5.2 Checking for Improper Usage

The ICD might be designed to detect, and perhaps report, a number of anomalous events such as (1) the receipt of an invalid initial connection request (e.g., attempted playback or forgery of a control message) and (2) the attempted transmission of a message with an indicated security level that is higher than allowed for the established connection. These features would add some complexity to the ICD, namely the need for detection capabilities and for reporting to the SC.

4.5.3 Augmenting the SC Monitoring Functions

The SC has little or no monitoring capability for the usage of a connection, and it is unlikely that the ICD can improve this situation to any significant extent. However, the ICD could augment the SC's capabilities in other ways such as by allowing the SC to "break" an existing connection under certain circumstances. Such a feature would have to be included in the initial design, if desired, and would also be affected by the type of communications net (e.g., direct dial or store-and-forward).

4.6 SECURITY ASSURANCE ASPECTS OF THE ICD

There is no reason to believe that the usage of ICD's will decrease the physical and procedural protection requirements from those of current crypto devices. The primary difference in this regard will be that the manual updates of private keys will not be required with as great a frequency as the manual updates of working keys. The procedural aspects of such changes would not necessarily differ.

The basic technology has changed considerably in recent years, and there are cost and reliability motivations to utilize the more recent techniques. Therefore, we can assume that the ICD development would be oriented towards the usage of microprocessor/ROM-logic*, and therefore could also utilize the proof-of-correctness techniques that were discussed in Section 3.6.

---

*There is a growing tendency towards the usage of ROM (Read Only Memory) program steps to replace hard-wired logic.

If ROM's were to be utilized in the ICD development, they might also serve as an ideal mechanism for the private keys. This might be particularly attractive for the case of programmable ROM's (PROM's).

4.7 OTHER ICD ASPECTS

The remaining items to be considered fall into two major categories: (1) those related to the cost/complexity issues of the ICD's and (2) those related to the communications network and the HOST-level control programs.

4.7.1 Cost/Complexity Issues

The ICD should be viewed as a set of components from which one can tailor a set of devices to meet differing operational and technological requirements. These differences are caused by differing requirements for:

- a dedicated terminal
- a multiplexed set of terminals
- a HOST system
- a Security Controller

and also by whether the communications network is:

- a set of leased lines
- a direct dial network
- a store-and-forward net
- a broadcast net

Since it is quite likely that networks will continue to involve a variety of needs, devices, and communication technologies, one should develop the ICD's with an appropriate degree of flexibility. However, the cost constraints will also differ such that one can not afford generality at the expense of raising the cost to all end users. Therefore, the modular (building block) approach seems most desirable. Referring back to Figure 4-5, we can assert that (1) there should be at least four versions of the interface to the data processing

equipment (for the four types of equipment listed above), (2) three versions of the crypto device (dedicated, multiplexed, and master), and (3) the four different network interfaces. Not all combinations of these three modules would be allowed (e.g., a master crypto device would not be attached to anything other than an SC), but the standardization of inter-module interfaces is still desirable. If current Large Scale Integration (LSI) technology is to be utilized in the development of ICD's, the need for volume production must be considered, and might swing the balance towards more commonality even if the resulting devices were more complex. Since the network and HOST interfaces are typically not standardized in any meaningful fashion, such commonality could only apply to the inner portion of the ICD, namely the cryptographic subsystem. If, on the other hand, hard-wired logic were to be utilized, there appears to be an economic advantage to simplifying the terminal ICD's whenever possible, at the expense of adding complexity to the HOST ICD's.

4.7.2 ICD-Level Control Programs

The network interface module of the ICD (as shown in Figure 4-5) should contain those control functions appropriate for the particular communications network. This would include the capability to automatically dial numbers in a direct dial network, or to create connections via protocol control messages in an ARPA-like net. All such network-dependent functions should be implemented in this module to ensure that the higher-level modules (e.g., the HOST interface) is generally usable.

5. NETWORK SECURITY AT THE COMMUNICATIONS NET LEVEL

In our introduction, we stressed that "the security problem of computer networking is not a communications problem, but another more sophisticated instance of multi-level computer operating system security" (quoted from Anderson AND-72). This was not intended to imply that there are no communications-related concerns at all, but rather that the major emphasis and concern should be placed at the higher levels of the network design. Having covered these higher level issues in earlier sections of the report, we will conclude with a "bottom-up" view as seen by the communications net, and will discuss the security-related aspects of this lowest level.

Our analysis was intended to cover a wide range of network architectures; although we have emphasized message-switching due to the increasing acceptance that it provides the best balance of availability, data rate, response time, error recovery and cost. The benefits (and limitations) of message-switching can be seen relative to other network communications technologies by considering their architectural differences. These differences manifest themselves along a number of dimensions, such that the observed variations are dependent upon the point-of-view or dimension being observed. For our purposes in evaluating communications net security, we will consider two major axes of the architecture as shown in Table 5-1. One axis lists the basic generic technologies of dedicated (point-to-point), circuit-switched, message-switched, and broadcast nets. This axis emphasizes the more structural aspects of the net, while the second axis considers the operational aspects of resource allocation, control, addressing, and fault-recovery.

Since any selected communications net would be operating on enciphered data for the end-to-end protection scheme, the security vulnerabilities of concern tend to be primarily in the area of denial of service. Such threats might come about via any of the operational aspects listed in Table 5-1, e.g., by modifying message addresses, by introducing line faults, or by exploitation of flaws in the control structure, and will be considered in more detail in Section 5.7.

Table 5-1. Comparison of Network Architectures

Generic Technology	Resource Allocation	Addressing	Control Structure	Fault Recovery
Dedicated point-to-point	Pre-allocated	Inherent in fixed structure.	Manual	Vulnerable to faults, manual recovery.
Circuit-switched	Allocated on demand for duration of dialog. Unallocated Resources tend to be pooled.	Established in pre-dialog set up of the connection.	Via dial mechanism (distributed).	Vulnerable, but can reconnect via a redial operation.
Message-switched	Allocated on a per-message basis. Unallocated resources tend to be pooled.	Explicitly included with each message.	Distributed and/or centralized. Dynamic control over messages, links, etc.	Highly redundant structure, with automatic recovery plus manual back-up.
Broadcast (incl. loop nets)	Completely pooled resource; allocated on a per-message basis.	All entities see all messages, but addressing is by explicit field.	Distributed and/or centralized. Tends to be more dynamic than any other scheme.	Highly pooled nature tends to be vulnerable to certain faults, although redundancy can be added.

In the following sections, we will consider the issues related to authentication, authorization, etc. following the same structure as in earlier volumes, even though some of these categories do not apply to the communications net to any significant extent.

5.1 IDENTIFICATION/AUTHENTICATION ISSUES

A common practice of authentication in the commercial environment is to utilize the "call back" scheme in which the caller identifies himself, and then hangs up and waits for the called party to call him back. This authentication is essentially by means of the correspondence between telephone numbers and physical locations, and provides little additional authentication above that of the ICD's, etc. The use of "call back" also requires that telecommunications equipment has the necessary hardware to place (as well as receive) calls. This additional unit also tends to be expensive (about $25/month rental for the Bell System model 801 unit for each such modem). Multiplexing this equipment is possible, as performed on the MERIT net (AUR-73), but may not be available in all localities. Automatic call placement equipment may be required for other operational reasons, but should not be considered as an authentication mechanism per se, since it would provide minimal security, (and might give a false sense of security to the operation).

One of the concerns related to denial of service is that malicious users might try to "tie-up" all of the incoming ports of a network resource. For a direct-dial network, this would involve making N calls, where N is the number of modems at the resource, and then trying to keep each such dialog open for as long as possible, e.g., by apparent slow typing of an identifier. The ICD-level test using an echoed message to establish that the two ends can communicate in enciphered form can provide a rapid, user-independent mechanism to ensure that such input-port hogging is minimized.

5.2 ACCESS AUTHORIZATION CHECKING

Little, if any, authorization checking can be performed at the communications net level, except to implement certain least-privilege limitations. For

example, if automatic dialing equipment were to be utilized, only the telephone numbers for authorized connections should be included. Such measures should be considered to be rather weak fire walls, and would provide value only for accident-induced security flaws.

5.3 ACCESS CONTROL; ESTABLISHMENT OF A CONNECTION

A requesting device (terminal, HOST, etc.) initially connects to the SC by addressing a "HELLO/id" message to it as discussed in Section 4. This initial connection is via the communications net, e.g., by dialing the phone number of the SC or by addressing a store-and-forward message to it depending on the communications net technology.* The considerations involved in this process are primarily operational, e.g., the time required for dialing, the cost and physical limitations of multiple input ports as opposed to multiplexing a single port, and the bandwidth required for the requestor-to-SC dialog. Considerations such as these will be discussed in the following paragraphs.

5.3.1 Initial Connection to the SC

Addressing the initial connection message to the SC would be via either a direct dial or an explicitly addressed message. The dialing time would be of the order of 20 seconds, which is small compared to the estimated total dialog between a requesting person and the SC (about one minute), but would be long compared to that of an automated (e.g., HOST computer) dialog.

5.3.2 Input Port Considerations at the SC

The SC could receive incoming requests via a set of physical ports or one multiplexed port depending on the communications net. Earlier estimates indicated the need to service of the order of 50 simultaneous requestors, which would result in considerable expense for telephone modems at the SC

*The dedicated (point-to-point) net is not considered further since it is not viable for the dynamic nature of requestor-to-SC and requestor-to-resource connections.

(e.g., $1000 per month based on 50 modems at $20/month each). A lesser number of automatic call modems would be required for SC initiated connections (e.g., to another SC).

Only the direct dial net requires that physically separate input ports be provided for each active user, since all other schemes perform message multiplexing, at some additional complexity within the SC.

5.3.3 Bandwidth Requirements for SC Dialogs

The requestor-to-SC dialog tends to involve relatively little information flow, and therefore is not affected appreciably by utilizing data rates above those provided by standard voice-grade communications. Only certain authentication methods (such as sending finger print scan data) would change this observation to any significant extent. Therefore, the available bandwidth will only be a minor factor in selecting of the requestor-to-SC communications.

5.3.4 Distributing the Working Keys

The manner in which the SC (master ICD) would distribute the working keys to the ICD's is highly dependent on the network technology used. For a direct dial net, either the relay or priming methods could be utilized with the following considerations affecting the decision.

- Use of the relay scheme: If the SC (master ICD) is to relay a connection message via a direct dial ICD, it must first send the message to the requesting ICD, at which time the connection to the SC would be broken and a new direct dial connection would be made to the resource. This requires that the requesting device be able to accept a telephone number sent from the SC and to then call this number (preferably automatically). The same requirements apply for relay messages via the resource.

  For a message-switching net, the key distribution is simplified to that of addressing a control message to the two ICD's, via either the relay or priming scheme.

- Use of the priming scheme: The SC could place a call to the resource and thereby deliver the matching key (as given to the requestor), at the expense of additional automatic call generation capability at the SC. One end of the working connection must still have call generation capabilities as in the first case.

5.3.5 Notification of Connection Status

If the SC (master ICD) is to be notified of the status of a connection that it attempted to establish, we must provide a mechanism by which a message can be returned to the SC. This is awkward, if not impossible, in the direct dial case since the ports are "tied-up" for the duration of the working connection, and hence can not be used to communicate status information to the SC. An auxiliary port could be utilized when a HOST is involved in a connection, but this would not handle the case of terminal-to-terminal connections. Message-oriented connections do not suffer any of these problems. Notifying the SC (master ICD) that a connection has been broken does not present any problem in either case, since at that point in time the ports are free.

5.3.6 Ability to Establish Priority Connections

Some priority control over access to the SC may be required, such that a high priority requestor is not locked-out or significantly delayed. Such problems are particularly of concern in those cases in which a physical resource is either permanently or temporarily dedicated to a user, such as the use of input ports for a direct dial scheme or use of the transmission ring in certain loop net designs. Priority override may be difficult in such cases since some preemptive capability is required. Multiplexed usage allows non-preemptive priorities since each "user" has control for a very brief period (milliseconds) and does not have control over who gets it next, i.e., can't hog the resource.

5.3.7 Establishing Connections in Process Addressed Nets

One added flexibility of broadcast nets (radio broadcast and loop nets) is that since all entities see each message, the physical location of a process can be made transparent to the message delivery vehicle. This feature is complicated considerably by the need for hardware-mechanized end-to-end protection, and for accountability and audit trail records.

5.4 USAGE OF A CONNECTION

Security aspects related to the usage of a connection include: (1) vulnerability to traffic analysis, (2) spoofability via play-back of recorded (and possibly modified) messages, and (3) denial of service. In each case, we assume that the "enemy" has physical access to the communications net, at least to the extent that monitoring devices could be inserted. In other cases, modification (or insertion) of messages would also be possible, and at least some lines could be damaged or destroyed.

5.4.1 Traffic Analysis

We have assumed that message leader information would be sent in the clear whenever any intermediate entities, such as message switches, need to determine how to route the flow. Such information might also be of interest to an enemy for traffic analysis purposes, e.g., to draw inferences from the occurrence, quantity and length of messages between two agencies. One obvious solution to this problem would be to encrypt the inter-node links on a link-by-link basis.

5.4.2 Spoofability

Depending on the encryption method, different spoofing methods can cause confusion, or at least errors, in communications. If a self-synchronizing method is utilized, then play-back of recorded messages is possible and must be countered by the usage of sequence numbers or time stamps. If these mechanisms are not provided, one might still be able to recognize a duplicate message, unless it had been modified. However, modification of the cipher text would result in the error propagation effect and thereby introduce a large number of erroneous bits, helping to ensure that the forgery was detected. (Check sums could also be utilized to detect changes in other encipherment schemes).

Spoofing threats can be countered by: (1) detecting modified messages by use of error checks on the clear text, (2) detecting the "replaying" of legitimate messages by the use of encrypted sequence numbers or time stamps, and (3) discarding any messages that do not meet these checks.

5.4.3 Denial of Service

Access to the physical communications net by an enemy results in a potential denial of service via the introduction of errors or, in the extreme, by cutting the wires or otherwise breaking the communications. Sufficient redundancy must be available to recover from such loss, preferably by some automatic mechanism.

5.4.4 Error and Flow Control

The communications network should provide some degree of control over errors, i.e., the ability to detect and recover from a variety of erroneous conditions. In addition, control over the acceptance of messages needs to be provided to avoid congestion within the net whenever resources are allocated dynamically, and particularly so when error conditions affect the need for resources such as line capacity, buffers, etc.

Error control typically consists of adding redundancy bits to each segment of a message (character, block, etc.) based on some algorithm, and then checking upon reception to ensure that the algorithm is still satisfied. For connections that involve more than one physical link between the two parties, we have an additional option, namely whether error control should be on a link-by-link basis or performed as an end-to-end check, or possibly that both types of checks should be made. The factors which affect the decision include the average message delay, message throughput, buffer storage, and the completeness of checking. Message delay is affected since an error on any link results in a retransmission of the message over all the links through which it must pass. Similarly, this need also reduces the message throughput. The message buffering requirement is affected in a more subtle fashion since a given message must be saved in one place or other, making it relatively insensitive to the choice between the two methods, but doubling the buffer requirements when a combination of the two methods is utilized.

Completeness of checking is enhanced by utilizing end-to-end checks since these will detect intra-node errors as well as inter-node (link) errors. For example, the ARPA net error checks on a link-by-link basis do not detect errors that occur within an IMP, and some additional checking has been added in selected cases such as for critical routing information (BBN-73B).

Flow control is a necessary, and surprisingly complex aspect to networking which can significantly affect the delay and throughput of a net as well as minimize (or cause) bottlenecks, deadlocks, and critical race conditions. Factors that are involved include the criteria for accepting new messages into the net, reserving storage space, and indications that messages have been successfully delivered. The topic is a subject in itself, and will not be pursued in detail here, since the ARPA net has provided an ideal forum for such considerations and has produced significant results in this area (CER-74, KAH-72).

5.5 SECURITY MONITORING

Very little security monitoring can be done at the communications net level other than attempting to ensure that denial of service threats do not result in a serious degradation to the network performance level. To perform this function, operational monitoring must be provided within the net, similar to the automated status reporting in the ARPA net (CRO-73). Such reports are readily available in a message-switching or broadcast net, but are probably not feasible in a direct dial net. The monitoring functions should be performed within the context of a network operation function (analogous to the ARPA net's Network Control Center), although there should be a means for this center to relay information to the Network Security Center whenever certain threshold conditions were exceeded.

5.6 SECURITY ASSURANCE

Considerable attention must be paid to ensure that the network does not have an "Achilles heel" vulnerability which could be utilized to "crash" the entire network. Such vulnerabilities can readily exist in nets that have been

designed to allow remote access to network switches for debugging or reloading purposes from a centralized control center. Such facilities are desirable, and possibly even necessary, for maintenance of a net, but must be very carefully controlled if the net is to be safe from accidentally or maliciously induced crashes.

If ARPA net IMP-like switches are utilized in a net, the necessary control over debugging, reloading, etc. might be by means of an attached HOST-level device which would control these operations. Communication of debug commands would then be via this HOST-level device and could be controlled by means of the normal HOST-level protection mechanisms. Depending on the extent to which this protection need be applied, this HOST-level entity might even be one of the software "fake-HOST's" (such as exist in the IMP's).

Severe network degradation can also be caused by errors within a switching node which propagate to neighboring switches in an infection-like manner that soon affects the entire net. This is more than academic speculation as was discovered in the ARPA net when IMP errors lead to defective routing information being passed between IMP's which thereby caused an eventual network crash. Extensive check-summing of routing data and programs was added to avoid such error-induced problems. However, in a network of the type that we are considering, such conditions could also be maliciously induced since the switches are not necessarily protected. Therefore, additional controls are required to detect abnormal routing updates in this environment.

5.7 MISCELLANEOUS ASPECTS

We will consider three separate topics in this final section; line control disciplines, TIP-related problems, and network architecture considerations.

5.7.1 Line Control Considerations

Two basically different classes of line control disciplines are utilized to "package" messages for delivery from a source to a destination; the character-oriented disciplines such as IBM's Binary Synchronous Communications, and

bit-oriented disciplines such as the Advanced Data Communications Control Protocol (ADCCP)*. A typical character-oriented discipline is shown in Figure 5-1(a) for comparison with the ADCCP as shown in part (b). The two schemes differ in several respects including:

- Message framing, which is by DLE STX and DLE ETX for the character-oriented discipline versus an 8-bit flag separator to indicate the beginning of one message and/or the end of the previous one.

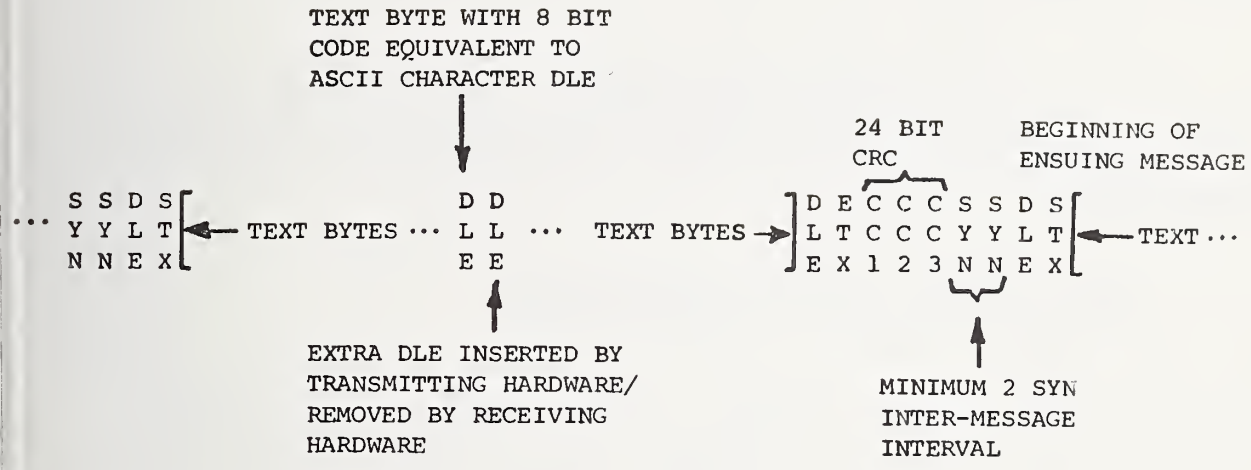
- Header control information, which is ad hoc for the former compared to being defined (in an open-ended manner) for the ADCCP.

- The information field, which is an arbitrary number of characters versus an arbitrary number of bits in length.

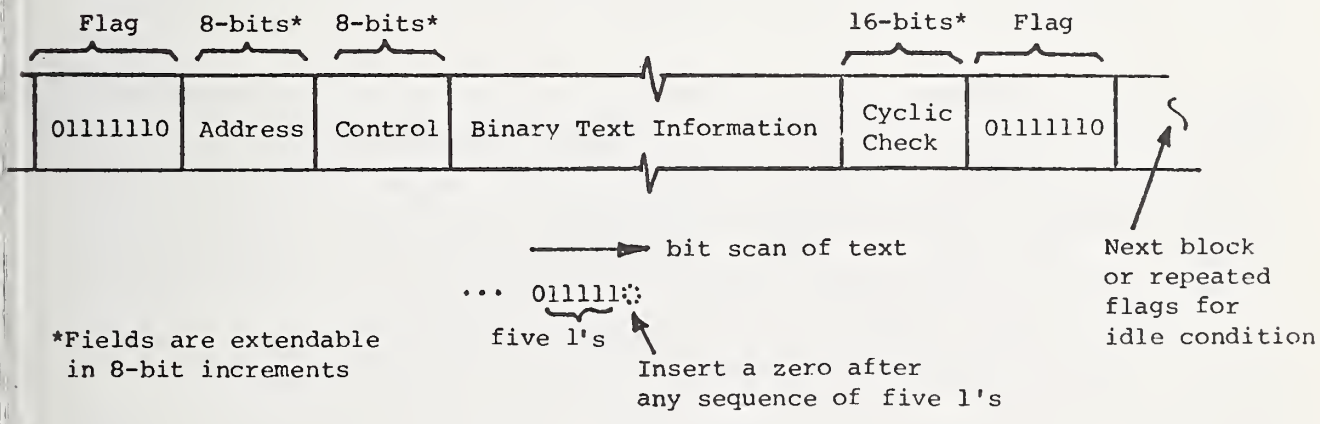
- The handling of transparent text, which is by DLE-doubling versus by "breaking up" accidental occurrences of the flag pattern.

- The need for character synchronization, which is a prerequisite to determining the DLE STX sequence, but is not required for the ADCCP framing scheme.

*"Line Control Procedures" by J. Gray, Nov. 1972, Proc of IEEE.



(a) Character-oriented line discipline (as used in the ARPA net).



(b) Bit-Oriented Line Discipline

Figure 5-1. Comparison of Character and Bit-Oriented Line Disciplines

Now that the ADCCP is becoming better known, future network designs will have to select between these two schemes rather than merely adopt a variation of the character-oriented discipline. Availability of an alternative has good and bad effects, e.g., it adds to the problems of integrating two or more networks, when different choices have been made for the individual nets.

There are no apparent security-related effects on the choice between the two methods, so the decision should be based on operational and compatibility considerations.

5.7.2 Network Terminal Handling Considerations

The Terminal Interface Processor (TIP) was introduced into the ARPA net as a combination IMP and terminal handler. It has provided a useful terminal interface, and has avoided the problems of using a large HOST merely as a "front end" to the HOST that is providing a service.* However, problems in its usage have indicated that this is not the proper way to interface terminals to a network. Adding security requirements to the net has even further emphasized the TIP-related problems.

Some of the TIP problems related to security are simply that it was not designed with security in mind, e.g., it does not perform any authentication or authorization checking, nor does it keep any audit trail information. These factors could be added to a design (or redesign), but a more fundamental TIP problem arises when end-to-end encipherment is required. The TIP requires that certain control information be "intermixed" with the messages from the terminal, and therefore would require that (1) only the message text be enciphered, and (2) that the enciphered text not have any accidental control

*Such dual HOST problems include those of reliability, extra costs, delays, etc.

character bit patterns in it. These are similar problems at the Network Control Program level, since the encipherment devices and NCP's are "crossed" as shown in Figure 5-2. This places the crypto device between the terminal and one NCP, and another between the two NCP's thereby eliminating the establishment of any straightforward, level-oriented control. A proper arrangement would clearly separate the levels.

5.7.3 Security Aspects of Different Network Architectures

Different network architectures are vulnerable to somewhat different security threats, although in practically all cases the basic threat is via denial of service. We shall consider seven different architectures, expanding most of our earlier four categories into subclasses for this discussion. These nets will be:

- Dedicated (point-to-point)
- Circuit-Switched
- Tree Structure
- Star Structure
- Multiply Connected
- Loop (ring) structure
- Radio Broadcast nets

5.7.3.1 A Dedicated Point-to-Point Net. A seemingly straightforward approach to controlling access between network entities is to directly interconnect all those devices authorized to communicate with each other, such that only those connections would exist in the net. If a given entity such as a HOST would change its security level during the day, an appropriate portion of its links would be enabled or disabled, giving some ability to adapt to change.

Several problems plague this simple scheme. In all but the smallest nets, the number of interconnection combinations quickly gets out of hand, since the number of meaningful connections tends to be a sizeable portion of the $n(n-1)$ different possible links connecting n entities. Also, implied connections

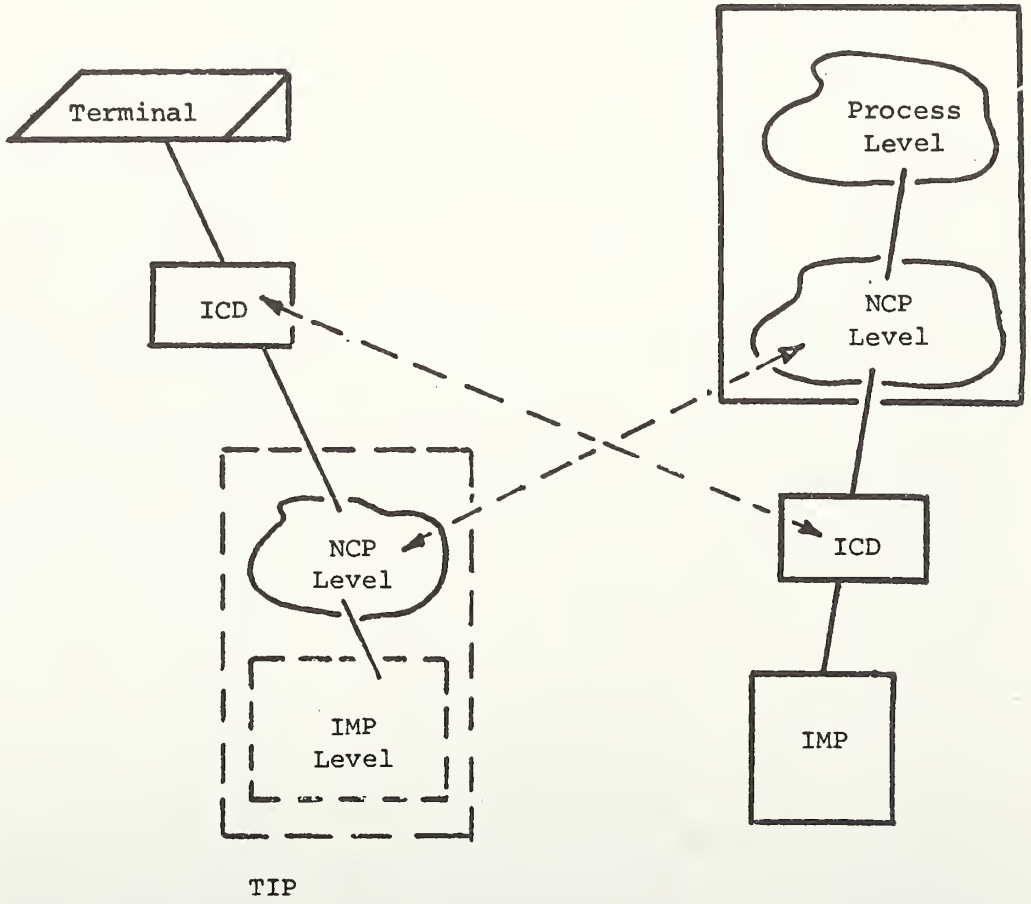


Figure 5-2. Difficulty Due to Crossing of Levels for ICD Usage with a TIP

via possible N-th party access tend to circumvent the careful isolation of the different entities, unless a hierarchical authorization scheme exists, which in itself is not necessarily proper security. Other "all or nothing" aspects to such an arrangement tend to violate our concepts of how network access should be determined and controlled. Therefore, the dedicated connection net represents one class of network structure, which is an interesting point on the spectrum of possibilities, but one that is too extreme for any general utility.

5.7.3.2 Circuit-Switched Network. The best example of a circuit-switched net is the direct dial telephone system, which of course can be utilized for data communications as well as voice. The principal problems in such usage are the limited bandwidth and the time required to establish the connection, while the primary advantages are its widespread existence and availability. The data rate of such lines is limited to 1200 bits/sec for asynchronous operation and of the order of 2000 bits/sec for synchronous operation.* Although the latter could be increased by the usage of automatically equalized modems, this equalization time would add even further to the approximately 20 seconds required for the dialing time. The lengthy time requirement to establish a connection typically means that it is maintained even for fairly lengthy idle periods. Even if the technology of connection establishment were to change, the minimum billing period would enter into the decision of when to break a connection and later re-establish it.

The direct distance dial net is particularly inefficient for the interactive user, who typically can never utilize the full line capabilities, and multiplexing of a dial connection is feasible only under conditions which tend to contradict the availability advantages of the direct dial net. A combination of direct dial and multiplexed point-to-point lines is often utilized, but borders on other combination nets such as a message-switched net with direct dial access.

*Direct dial service at 50K bits/sec. is available in selected major cities (Dataphone-50).

The security-related aspects of the direct dial net are largely related to its impact on the cryptographic equipment, and in particular, whether multiplexed cryto devices are economically advantageous. For example, if individual direct dial lines are brought to a HOST computer, they would then have to be multiplexed prior to entering the crypto device. Much of the flexibility of "addressed" multiplexing e.g., by the use of message headers, is therefore lost, and many of the physical port constraints begin to show up on the design of the multiplexed crypto device. Handling of the large number of input lines, connectors, etc., may also grow beyond expectations for such usage.

Some minor security advantages of the direct dial net are its "call-back" feature (as discussed in Section 5.1) and the difficulty that an enemy would have in performing any meaningful traffic analysis. There is also a large (apparent) redundancy in the direct dial net, but there are probably a number of sensitive points which would be very vulnerable to sabotage and would thereby sever a large portion of the user community from the net. In addition, malicious users might "tie-up" all of the input ports, thereby denying service to legitimate users.

5.7.3.3 Tree-Structure Nets (Message-Switched). The tree structure is occasionally utilized for networks when its relatively low line cost, and hierarchical organization match the needs of the network community, and when its high vulnerability to loss of any link is acceptable (or correctable by back-up methods). This structure posses some interesting problems in terms of where the ICD's would be located, and where the SC(s) would fit in the structure, but these considerations were not pursued due to the limited utility of the tree net in the applications of concern which require high availability of basically non-hierarchical resources.

5.7.3.4 Star Nets (Message-Switched). The star topology is also very vulnerable to loss of components, particularly the central switch, and to a lesser extent, to any link since that loss would sever one entity from the net. Line costs would also be high if the network entities are separated by inter-city distances

and the operational performance can degrade rapidly when a large number of small messages must be handled concurrently (i.e., switch saturation). There are some minor positive factors as well, such as the convenient spot for monitoring operations, namely the central node. However, since it is vulnerable to overload, adequate monitoring may not be feasible.

Denial of service is the greatest security threat of the star network, particularly due to its exceptional vulnerability to the loss of components or message flooding as discussed above.

5.7.3.5 Multiple Connected Message-Switched Nets. The reliability/availability disadvantages of star and tree nets can be overcome by adding redundant links between the nodes. The particular structure of the net can then become independent of any predefined topology, and instead can be based on expected traffic loads and geographical locations. The ARPA network is the prime example of this type of network, although other nets should also be included in this class.

The major security related advantage of the multiply-connected message-switched net is its high resistance to errors and/or malicious damage. This flexibility is, at the same time, its only apparent security disadvantage since complication tends to breed exploitable combinations of events and circumstances. This subjective observation is not an indictment against message-switching; it is merely a word of caution in the usage of, what appears to be, the best available data communication technology available.

5.7.3.6 Loop (Ring) Networks. A loop network is one in which all resources share a common communication bus which is closed upon itself forming a ring. Because of its structure, each transmission returns to the originator, delayed by some amount dependent upon the propagation delays of the loop components. If this delay is small (less than a message transmission length), the loop is typically controlled by a form of time multiplexing in which each entity gets a turn at "owning" the loop for its transmission. If the total loop propagation time exceeds the transmission time for two or more messages, one can take

advantage of the storage capability (as in a delay line) and can thereby multiplex several messages on the loop at a given time.

One of the most attractive aspects of a loop net is based on the fact that each entity on the ring sees every message as it goes by, and therefore one can address messages to a given process (instead of to a physical processor). The only requirement is that each interface be able to match process names (from message addressing) with a list of current processes which it contains. Other advantages that have been anticipated, (and perhaps achieved) are based on the expected low cost of the interfaces between the devices and the net, and the simple communications technology which utilizes only digital devices (similar to the telephone T1 carrier).

The single loop topology is inherently vulnerable to the loss of a line segment. Although back-up paths can be added, the increase in complexity and added line costs tend to detract from the attractiveness of the loop except for well controlled, local environments. Therefore, the loop net would seem to be an appropriate candidate for a local subnet, but not for the global net to interconnect such subnets.

Other security-related aspects include some increased vulnerability to traffic analysis, since all messages "go by" any given spot on the ring. However, message leaders could be encrypted (with a common key) to avoid this problem. More serious considerations are the difficulty in handling mobile processes by use of physically attached cryptographic equipment and the difficulty in implementing a priority override scheme.

5.7.3.7 Radio Broadcast Nets. Like loop nets, the radio broadcast networks allow each entity to see every message, and to contend for a common communication resource. However, they differ in a number of other important operational aspects such as (1) the ability to have separate, asymmetric data rates for the two sides of a dialog, (2) the way in which requestors attempt to gain control of the communications media, and (3) the ease of "tapping" the communications. Since we assume an open communications net in all cases,

this last difference is academic. However, the second difference can be an important one since it is typically handled by a scheme that is basically "transmit and see if it gets through." The resulting conflicts due to overlapped transmissions reduce the theoretical (average) capacity to about 18% of its actual capacity, although this can be doubled by the use of "slotted line" methods in which transmissions are only initiated at fixed intervals, thereby reducing the conflict probability.

A severe restraint on the usage of broadcast nets is the lack of available frequency spectrum, and to a lesser extent, the geographical coverage problems. (Satellites have favorably impacted the latter consideration).

Security vulnerabilities tend to be related to the highly centralized nature of the communication resource, which makes the net very vulnerable to its loss, thereby introducing denial of service threats. Jamming might also be a consideration, although anti-jamming methods are undoubtedly available to circumvent this problem.

6. SUMMARY OF RESULTS

The computer network security problem is not merely a communications problem, but rather a complex set of problems that are due to the multi-system nature of nets. In effect, the network environment adds this new dimension to the multi-user, multi-resource problems of a single system, and therefore requires additional security controls beyond those of single systems. The most viable mechanisms proposed to date are the Security Controller and Intelligent Cryptographic Device, which were originally described by Branstad (BRA-72) in a paper which formed the starting point for our investigation. Initially, this approach looked very promising, and as our investigation proceeded, it appeared even more appropriate as a basic solution to the problems of network security. The deeper we probed, the more convinced we became that this approach does represent the correct solution, that the necessary technology is available, and that such mechanisms are needed now, and will be needed even more in future years as networks become increasingly prevalent.

The problems of network security are many faceted, and therefore presented us both with technical problems and with problems in how to organize this material for presentation. We chose to consider the network as consisting of several levels, and proceeded in a top-down analysis involving: (1) the policy and requirements issues, (2) the HOST/SC systems, (3) the ICD's, and (4) the communications network. Within each level, we considered the issues related to authentication, authorization, connection establishment, connection usage, security monitoring and security assurance. While this systematic scan of the network security considerations resulted in a large number of design issues and tradeoffs, the more salient portions of the analysis are felt to be those related to:

- Determining the security vulnerabilities of a computer network and defining requirements to ensure network security.
- The controlled establishment of connections via the SC.
- The rigid adherence to maintaining the appropriate separation of protocol layers (at physical and abstract levels) in the design.

- The preliminary design of the SC including its control program and duplexed hardware considerations.
- Error control considerations at each level.
- The network interface considerations for the user, HOST computers, and ICD's.
- Establishing the basic requirements for the ICD (e.g., control primitives, relay capability, buffering, multiplexing, and error control).
- An analysis of the various communication net architectures, and their security strengths and vulnerabilities.
- Defining auxiliary mechanisms for a secure net (Net Security Center) and for networks of networks (Gateways).

In addition to technical considerations, we treated procedural, and economic aspects whenever possible. The cost of the SC and ICD's is difficult to estimate due to the large number of unknowns related to quantities, packaging considerations, testing, etc., but no unforeseen expenses were uncovered in the investigation. The performance impact due to security is also very dependent upon operational considerations, but is estimated to be very small. Some improvements may occur where the high level of system integrity reflects in improved reliability and availability, and some degradation may occur due to the security overhead.

In conclusion, the SC/ICD approach will provide the necessary control mechanisms to handle the complications of the network environment, and to provide a viable and evolutionary approach to achieving this goal in both existing and future networks.

BIBLIOGRAPHY

AND-72 Anderson, J. P., "Computer Security Technology Planning Study," ESD-TR-73-51, Vol I & II, Oct. 1972.

AUE-74 Auerbach, K., "An Analysis of WWMCCS ADP Security; Vol 2 - Data Communication Networks," SDC Tech Memo TM-WD-5733/004/00, March 1974

AUP-73 Aupperle, E. M., "MERIT Network Re-examined," COMPCON 73, Feb. 1973 pp. 25-29.

BAR-64 Baran, P., "On Distributed Communications: Vol IX, Security, Secrecy, and Tamper-Free Considerations," RAND Memorandum, RM-3765-PR, August 1964.

BBN-74A Bolt Beranek and Newman, "Interface Message Processors for the ARPA Computer Network," Rept. 2717, Jan. 1974.

BBN-74B Bolt Beranek and Newman, "Interface Message Processors for the ARPA Computer Network," Rept. 2816, April 1974.

BLA-73 Blanc, R. P., et al, "Annotated Bibliography of the Literature on Resource Sharing Computer Networks," Nat Bur of Stds, NBS Spec. Pub. 384, Sept. 1973. (See WOO-76)

BLA-74 Blanc, R. P., "Review of Computer Networking Technology," NBS Tech Note #804, Jan. 1974.

BOU-73 Bouknight, W. J., et al, "The ARPA Network Terminal System - A New Approach to Network Access," Third Data Communications Symposium (IEEE and ACM), Nov. 1973, pp. 73-79.

BRA-73 Branstad, D. K., "Security Aspects of Computer Networks," AIAA Computer Network Conference, April 1973.

BUS-74 Bushkin, A. A., "A Framework for Computer Security," SDC Tech Memo TM-WD-5733, March 1974.

CER-74 Cerf, V. G., "An Assessment of ARPANET Protocols," ARPA Net Working Group Note #635, April 1974.

CRA-73 Craig, D., "Computer Networks; A Bibliography with Abstracts," Nat. Tech Info. Center, NTIS-WIN-73-087, COM-73-11977/8WC, Nov. 1973.

CRO-72 Crocker, S. D., et al, "Function-Oriented Protocols for the ARPA Computer Network," SJCC Proceedings, 1972, pp 271-279.

CRO-73 Crowther, W. et al, "Reliability Issues in the ARPA Network," Third Data Communications Symposium (IEEE and ACM), Nov. 1973, pp 159-160.

DAV-73 Davies, D. W., and Barber, D., "Communication Networks for Computers," Wiley, 1973.

FAR-72 Farber, D. J., "Networks: An Introduction," Datamation, April 1972, pp 36-39.

FAR-72A Farr, M.A.L., et al, "Security for Computer Systems," Pub. by National Comp. Centre, Ltd., London.

FAR-73 Farber, D. J., et al, "The Distributed Computer System," Seventh Annual IEEE Comp. Soc. Int. Conf., March 1973.

FAR-74 Farber, D. J. and Vittal, J., "Extendability Considerations in the Design of the Distr. Comp. Sys., UC Irvine Technical Paper, 1974.

GAR-73 Garwick, J. "Security Controller Design" and "The Security Profiles of Users and Files and their use for Access Control," SDC Tech Memo TM-5211 Vols 4 and 7 respectively, August 1973.

GAR-74 Garwick, J., "Programming the Security Controller," SDC Tech Memo TM-5346/000, June 1974.

GRY-74 Grycner, H., "Fault Detection in the Sec ·ity Controller," SDC Tech Memo TM 5346/001, August 1974.

HAR-73 Harrison, A., "Computer Information Security and Protection: A Bibliography with Abstracts," NTIS-WIN-73-052, Oct. 1973.

HAS-73 Hassing, T. E., et al, "A Loop Network for General Purpose Data Communications in a Heterogeneous World," Third Data Communications Symposium (IEEE and ACM), Nov. 1973, pp 88-96.

HIC-70 Hicken, G. M., "Information Network of Computers," in Fourth Gen. Computer User Req'mts and Transition, F. Gruenberger (Ed), Prentice-Hall, 1970, pp 31-58.

HIC-71 Hicken, G. M., "Experience with an Information Network," Fifth Annual IEEE Computer Society Conf., Sept. 1971, pp 169-170.

JAC-69 Jackson, P. E. and Stubbs, C. D., "A Study of Multi-Access Computer Communications," 1969 SJCC, pp 491-504.

JON-73 Jones, A. K., "Protection Structures," PhD Thesis, Carnegie-Mellon Univ., 1973.

KAT-73 Katzan, H. Jr., "Computer Data Security," Van Nostrand Reinhold Co., 1973.

KAU-74 Kaufman, D. J., "Access Control Information in a Computer Network, SDC Tech Memo TM-5346/002, August 1974.

LAM-69 Lampson, B. W., "Dynamic Protection Structures," 1969 FJCC, pp 27-38.

LIP-72 Lipner, S. B., "SATIN IV Computer Security," MITRE, MWP-4445, Sept. 1972.

LIP-73 Lipner, S. B., "Computer Security Research and Development Requirements," MITRE, MTP-142, Feb. 1973.

LIP-74 Lipner, S. B., "A Minicomputer Security Control System," MITRE, MTP-151, Feb. 1974. (Also IEEE Comp. Society Int. Conf., San Francisco, Feb. 1974.)

LOM-72 Loomis, D. C., "Ring Communication Protocols," UC Irvine DCS Project Note #46-A, May 1972.

LOR-73 Loret, B. J., "Prototype Worldwide Military Command and Control System Intercomputer Network," AIAA Computer Net Conf., Huntsville, Alabama, April 1973.

LUP-73 Lupton, W. L., "A Study of Computer Based Data Security Techniques," Naval Postgrad-School Thesis, (AD 765 677) June 1973.)

MET-72 Metcalfe, R. M., "Strategies for Operating Systems in Computer Networks," Proc. of ACM 1972 Nat. Conf., pp 278-281.

MOL-70 Molho, L. M., "Hardware Aspects of Secure Computing," 1970 SJCC, pp 135-141.

NEU-73A Neumann, A. J., "Review of Network Management Problems and Issues," NBS Tech Note #795, Oct. 1973.

NEU-73B Neumann, A. J., "User Procedures Standardization for Network Access," NBS Tech Note #799, Oct. 1973.

NEU-73C Neumann, A. J., "Network User Information Support," NBS Tech Note #802, Dec. 1973.

PET-71 Peterson, J. and Veit, S., "Survey of Computer Networks," MITRE, MTP-357, Sept. 1971.

PET-74 Peters, B., "A Program for the Development of ADP Security for WMMCCS," SDC Tech Memo TM-WD-5734/000, March 1974.

POP-74A Popek, G. J., "A Principle of Kernel Design," 1974 NCC, pp 977-978.

POP-74B Popek, G. J. and Kline, C. S., "Verifiable Secure Operating System Software," 1974 NCC, pp 145-151.

POS-74 Postel, J. B., "A Graph Model Analysis of Computer Communication Protocols" PhD Dissertation, UCLA-ENG-7410, Jan. 1974.

POU-73 Pouzin, L., "Presentation and Major Design Aspects of the Cyclades Computer Network," Third Data Communications Symposium (IEEE and ACM), Nov. 1973, pp 80-87.

PYK-73 Pyke, T. N., "Some Technical Considerations for Improved Service to Computer Network Users," Proc. of COMPCON 1973, pp 53-55.

ROB-70 Roberts, L. G. and Wessler, B. D., "Computer Network Development to Achieve Resource Sharing," 1970 SJCC, May 1970, pp 543-549.

ROB-73 Roberts, L. G., "Network Rationale: A 5-year Re-evaluation," COMPCON 1973, Feb. 1973, pp 3-5.

ROW-74 Rowe, L. A. et al, "Software Methods for Achieving Fail-Soft Behavior in the Distributed Computing System," UC Irvine Technical Paper.

RUS-72 Rustin, R. (Ed), "Computer Networks," Third Courant Computer Science Symposium (1970), Prentice-Hall, 1972.

SAV-67 "Some Simple Self-Synchronizing Digital Data Scramblers," Bell Sys. Tech No., Feb 1967, pp 449-487.

TAS-73A Tasker, P. S. and Bell, D. E., "Design and Certification Approach: Secure Communications Processor," MITRE Tech Rept., MTR-2436, June 1973.

TAS-73B Tasker, P. S., "Design of a Secure Communications Processor: Central Processor," MITRE Tech. Rept., MTR-2439-3, June 1973.

TOR-73 Torrieri, D. J., "Word Error Rates in Cryptographic Ensembles," Naval Res. Lab, NRL Rept. 7616, (AD 769 458) October 1973.

TYM-71 Tymes, L. "Tymnet - A Terminal Oriented Communications Network," 1971 SJCC, May 1971, pp 211-216.

WAL-72 Walden, D. C., "A System for Interprocess Communication in a Resource Sharing Computer Network," Com. of ACM, April 1972, pp 221-230.

WIN-74 Winkler, S. and Danner, L., "Data Security in the Computer Communications Environment," Computer (IEEE Comp. Soc.), Feb. 1974, pp 23-31.

WIS-69 Weissman, C., "Security Controls in the ADEPT-50 Time Sharing System," 1969 FJCC, pp 119-133.

WOO-76 Wood, H. M., et al, "Annotated Bibliography of the Literature on Resource Sharing Computer Networks," Nat. Bur. of Stds., NBS Spec. Pub. 384, revised Sept. 1976.

WUL-73 Wulf, W. et al, "Hydra: The Kernel of a Multiprocessor Operating System," Carnegie-Mellon Univ., AD 762 514, June 1973.

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS SP 500-21, Vol. 1	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

COMPUTER SCIENCE & TECHNOLOGY
Design Alternatives For Computer Network Security

5. Publication Date January 1978

6. Performing Organization Code

7. AUTHOR(S) Dennis K. Branstad, Editor

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Gerald D. Cole, Author
System Development Corporation
2500 Colorado Avenue
Santa Monica, CA 90406

10. Project/Task/Work Unit No. 6401112

11. Contract/Grant No. NBS 5-35934

12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP)

Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

13. Type of Report & Period Covered Initial

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 77-608320

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

The security problems associated with a network of computers are an extension of those of stand-alone computer systems, but require additional security controls due to the distributed and autonomous nature of the network components. The purpose of this investigation was to generate a pre-development specification for such security mechanisms by determining the issues and tradeoffs related to network security over a broad range of network applications, topologies and communications technologies.

The approach which was taken was that of utilizing a dedicated network Security Controller (minicomputer) for checking the authentication of requestors, and, to some extent, for authorization checking as well. The enforcement of the Security Controller functions would be by means of Intelligent Cryptographic Devices, which could be remotely keyed by the Security Controller when a requested communication was authorized. The Intelligent Cryptographic Device would incorporate the National Bureau of Standards Data Encryption Standard algorithm.

The investigation showed that this approach is a viable solution to the network security problems of a large class of computer networks, and that such security mechanisms should be developed for operational usage.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Access control; authentication; communication; computer networks; cryptography; encryption; security.

18. AVAILABILITY

[X] Unlimited

[ ] For Official Distribution. Do Not Release to NTIS

[X] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13.10:500-21, Vol. 1

[ ] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151

19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED

21. NO. OF PAGES 173

20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED

22. Price $6.00 per set

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SCIENCE & TECHNOLOGY

Superintendent of Documents,
Government Printing Office,
Washington, D. C. 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name _____

Company _____

Address _____

City _____ State _____ Zip Code _____

(Notification key N-503)

☆ U.S. GOVERNMENT PRINTING OFFICE : 1978 O—253-748

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology, and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent NBS publications in NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $17.00; foreign $21.25. Single copy, $3.00 domestic; $3.75 foreign.

Note: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

DIMENSIONS/NBS

This monthly magazine is published to inform scientists, engineers, businessmen, industry, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on the work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing.

Annual subscription: Domestic, $12.50; Foreign $15.65.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a world-wide program coordinated by NBS. Program under authority of National Standard Data Act (Public Law 90-396).

NOTE: At present the principal publication outlet for these data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St. N.W., Wash., D.C. 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The purpose of the standards is to establish nationally recognized requirements for products, and to provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.*

Order **following** *NBS publications—NBSIR's and FIPS from the National Technical Information Services, Springfield, Va. 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services (Springfield, Va. 22161) in paper copy or microfiche form.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

Cryogenic Data Center Current Awareness Service. A literature survey issued biweekly. Annual subscription: Domestic, $25.00; Foreign, $30.00.

Liquified Natural Gas. A literature survey issued quarterly. Annual subscription: $20.00.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $30.00. Send subscription orders and remittances for the preceding bibliographic services to National Bureau of Standards, Cryogenic Data Center (275.02) Boulder, Colorado 80302.

J.S. DEPARTMENT OF COMMERCE
Vational Bureau of Standards
Vashington, O.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215

U.S.MAIL

SPECIAL FOURTH-CLASS RATE
BOOK